THE AI TOOLCHAIN

NO. 034

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 22, 2026 · EVERY DAY

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

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | **EVERYTHING**

Every tool in the issue, in full.

all 571 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 57 of 59 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: Skyvern can keep browser state so automations stop redoing logins, Weights & Biases Weave gives agent

vim ? help

teams a turn-by-turn production view, and Fireworks AI can pick validated deployment shapes so serving config is less hand-tuned.

BUILD

Run browser agents without replaying the login ceremony every time

Persistent browser sessions and reusable profiles let an automation carry authenticated state across runs. That removes brittle preambles whose only job was to get past sign-in before the real task starts.

Skyvern

OPERATE

Debug agent behavior at the conversation turn, not just the trace span

Weave added an Agents view with conversations, spans, signals, and comparison screens; Braintrust added scheduled investigations for recurring failure modes; LangSmith added trajectory/turn inspection and export. Together they make failures easier to cluster and replay from real runs instead of asking engineers to manually read raw traces.

Weights & Biases Weave · Braintrust · LangSmith

DEPLOY

Ship model serving with fewer bespoke capacity guesses

Fireworks AI added deployment-shape controls that can auto-select a validated shape, while vLLM focused on faster restarts and cold starts through weight caching and faster CUDA graph capture. The practical gain is less hand-picking of serving shape and less downtime while workers come back online.

Fireworks AI · vLLM

Does **GPT-5.5** read better? 👍 👎

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.91 NOTES ↗ RANK

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

> xalgorix shipped a run of token-economy features this window: two flag-gated token-saver toggles, a full token-usage observability endpoint and dashboard tab, new Mistral zai-glm-latest (GLM 5.3) model support, and durable scan-state persistence across server restarts.

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

01 Token usage observability endpoint and dashboard NEW

100

Adds a `GET /api/scans/{id}/token-usage` endpoint returning per-request token attribution for running and completed scans, with cache metrics reported as null when the provider does not return them. Persists compact metadata-only `token-usage.jsonl` and `token-usage.json` files under the scan directory that survive scan completion, server restarts, and resumption. Adds context-growth analysis (first/last/median/p95/max prompt tokens, prompt growth ratio, cumulative prompt tokens, resend-amplification heuristic), a breakdown by agent (root, authz-logic, injection-serverside, client-source, verifier) and by active-context source (system, assistant history, tool results, skill results, other user history), plus recovery-loop stats (retries, no-tool, malformed-tool, finish-rejection). Logs a `[token-analysis]` summary at scan completion and a redacted one-time MiniMax usage-structure diagnostic for streaming and non-streaming calls, and adds a new Token usage tab in the scan dashboard with a prompt-tokens-per-request chart filterable by agent.

— Names the exact endpoint, file formats, breakdown dimensions and UI tab. v4.6.89

02 Token-saver flags for bounded context and role-scoped tools

NEW

95

Adds `XALGORIX_BOUNDED_CONTEXT` (dashboard toggle `Bounded working context (token saver)`) and `XALGORIX_ROLE_SCOPED_TOOLS` (toggle `Role-scoped specialist tools (token saver)`) under Settings → Environment (LLM category). Bounded context converts aged tool-result messages into retrieval stubs archived to disk under `<ScanDir>/tool-outputs/` and fetchable via the new `read_tool_output` tool, cutting input tokens 40–60%, tunable via `Bounded context: active window` (default 8) and Bounded context: archive threshold (bytes) (default 1500); role-scoped tools withholds role-foreign tool docs from specialist prompts for an additional 10–20% reduction. Both flags hot-apply to the live runtime config on save with no restart required.

Enable bounded context to cut input-token costs by 40–60% on long scans where tool output accumulates across many steps.

> 📍 In the dashboard, go to Settings → Environment (LLM category), enable Bounded working context (token saver), set `Bounded context: active window` to 8 and Bounded context: archive threshold (bytes) to 1500, then click Save — no restart required.

— Names exact env vars, UI path, tunables and defaults with a runnable example. v4.6.91 v4.6.90

---

03 **Durable scan state persistence across restarts** NEW

70

Persists in-flight `iterations`, `total_tokens`, and `tool_calls` in durable `QueueState` files so scan state survives server restarts, flushes live instance metrics atomically on `SIGINT` / `SIGTERM` daemon shutdowns, and aggregates completed and running child sub-scan tokens, iterations, and tool calls into wildcard parent scan records.

— Names the persisted fields and shutdown signals but no user-facing command. v4.6.85

---

04 **Mistral zai-glm-latest (GLM 5.3) model support** NEW

65

Adds support for `zai-glm-latest` (GLM 5.3) via the Mistral integration, enabling OpenAI-compatible typed-part content arrays (e.g. `[{"type":"text","text":"..."}]`) in API responses, and handles streaming mixed deltas and thinking blocks with full token accounting for Mistral models.

— Names the model and response format but no direct usage steps. v4.6.85

THINNER COVERAGE BELOW

---

05 **Deterministic tool schema serialization** IMPROVED

45

Tool schema serialization is now deterministically sorted, fixing previously randomized per-build ordering that defeated provider prefix-cache stability.

— Explains the fix but gives no reader-facing action or config. v4.6.90

---

v4.6.91 NOTES ↗ LATEST STABLE

xalgorix v4.6.91 exposes token-saver flags `XALGORIX_BOUNDED_CONTEXT` and `XALGORIX_ROLE_SCOPED_TOOLS` in the dashboard with no restart required.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.91
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.91
```

└──▷ HOW TO FIND IT

Enable bounded context to cut input-token costs by 40–60% on long scans where tool output accumulates across many steps.

📍 In the dashboard, go to Settings → Environment (LLM category), enable Bounded working context (token saver), set `Bounded context: active window` to 8 and Bounded context: archive threshold (bytes) to 1500, then click Save — no restart required.

Adds `XALGORIX_BOUNDED_CONTEXT` (toggle: Bounded working context (token saver)) to the dashboard under Settings → Environment (LLM category), reducing input tokens by 40–60% by converting aged tool-result messages to retrieval stubs archived to disk and fetchable via `read_tool_output`; tunable via `Bounded context: active window` (default 8) and Bounded context: archive threshold (bytes) (default 1500).

Adds `XALGORIX_ROLE_SCOPED_TOOLS` (toggle: Role-scoped specialist tools (token saver)) to the dashboard under Settings → Environment (LLM category), delivering an additional 10–20% input-token reduction by withholding role-foreign tool docs from specialist prompts.

Hot-applies token-saver flag changes to the live runtime config on save, so new scans pick up updated settings immediately without a service restart.

v4.6.90 NOTES ↗ STABLE

xalgorix v4.6.90 adds two flag-gated token-optimization levers targeting 40–60% input-token reduction for scanner agents.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.90
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.90
```

Adds `XALGORIX_BOUNDED_CONTEXT=1` env var: aged tool-result messages become compact retrieval stubs while complete raw output is preserved byte-identical under

`<ScanDir>/tool-outputs/` and fetched on demand via the new `read_tool_output` tool, targeting 40–60% input-token reduction.

Tool schema serialization is now deterministically sorted, fixing randomized-per-build ordering that defeated provider prefix-cache stability.

v4.6.89 NOTES ↗ STABLE

xalgorix v4.6.89 adds token-attribution observability via a new `GET /api/scans/{id}/token-usage` endpoint and scan dashboard tab.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.89
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.89
```

Adds `GET /api/scans/{id}/token-usage` endpoint returning per-request token attribution for running and completed scans, with cache metrics reported as null when the provider does not return them.

Adds compact metadata-only persistence as `token-usage.jsonl` and `token-usage.json` under the scan directory, surviving scan completion, server restarts, and scan resumption.

Adds context-growth analysis per scan: first/last/median/p95/max prompt tokens, prompt growth ratio, cumulative prompt tokens, and a resend-amplification heuristic.

Adds token usage breakdown by agent (root, authz-logic, injection-serverside, client-source, verifier) and by active-context source (system, assistant history, tool results, skill results, other user history), plus recovery-loop stats (retries, no-tool, malformed-tool, finish-rejection).

Adds a `[token-analysis]` summary logged at scan completion and a redacted one-time MiniMax usage-structure diagnostic (field names only) for streaming and non-streaming calls.

+1 more

v4.6.85 NOTES ↗ STABLE

xalgorix v4.6.85 adds Mistral zai-glm-latest (GLM 5.3) support and durable scan state persistence across server restarts.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.85
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.85
```

Adds support for `zai-glm-latest` (GLM 5.3) via Mistral integration, enabling use of OpenAI-compatible typed-part content arrays (e.g. `[{"type":"text","text":"..."}]`) in API responses.

Persists in-flight `iterations`, `total_tokens`, and `tool_calls` in durable `QueueState` files so scan state survives server restarts.

Flushes live instance metrics atomically on `SIGINT` / `SIGTERM` daemon shutdowns.

Aggregates completed and running child sub-scan tokens, iterations, and tool calls into wildcard parent scan records.

Handles streaming mixed deltas and thinking blocks with full token accounting for Mistral models.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

Warp

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗ RANK

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

> Warp's latest update adds a CLI subcommand for creating container registry secrets, native shell completions, fzf/atuin triggering, and a cluster of team/workspace membership management improvements, alongside smaller polish to Agent Mode and command history search.

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

01 Container registry secret creation via CLI NEW 75

Adds the `oz secret create docker-registry` CLI subcommand to create container registry credential secrets, useful for authenticating against private Docker registries.

Create a container registry credential secret for a private Docker registry from the CLI.

```
$ oz secret create docker-registry
```

— Names exact subcommand and use case changelog-20260922-590604e6

02 Native shell completions in terminal input NEW 70

Warp now adds native shell completions alongside its built-in completion specs, falling back to native completions when no built-in suggestions exist. Configurable in Settings > Features > Terminal Input, where the option can be enabled as primary, kept as fallback, or disabled.

Enable native shell completions as the primary suggestion source instead of the fallback, or disable them entirely, via Warp's terminal input settings.

> 📍 In Warp, go to Settings › Features › Terminal Input and adjust the native shell completions option.

— Behavior and exact settings path given changelog-20260922-590604e6

03 fzf and atuin triggering inside Warp NEW 60

Supports triggering `fzf` (via `ctrl-r` or `ctrl-t`) and `atuin` from within Warp when configured with your shell.

— Names exact keybindings and tools changelog-20260922-590604e6

THINNER COVERAGE BELOW

04 **Windows Materials window backdrop** NEW 55

Adds support for Windows Materials as the backdrop of Warp windows, available in Settings > Appearance > Window > Backdrop.

Apply a Windows Materials backdrop to Warp windows for a modern glass-style appearance.

> 📍 In Warp on Windows, go to Settings › Appearance › Window › Backdrop and select a Material option.

— Exact settings navigation provided `changelog-20260922-590604e6`

---

05 **Team and workspace membership management** NEW 50

Workspace admins can now remove members from a workspace directly from team settings; native workspace members can leave a team without losing workspace access; and native workspace users can browse and join another open team from Teams settings.

— Names Teams settings surface, three behaviors listed `changelog-20260922-590604e6`

---

06 **Immediate interrupt on shared Agent Mode stop** IMPROVED 45

Stopping a shared Agent Mode conversation from a viewer or via conversation steering now interrupts the agent's running command immediately.

— Describes trigger and effect but no UI path `changelog-20260922-590604e6`

---

07 **Improved command history search ranking** IMPROVED 30

Improves command history search ranking using match quality and usage signals.

— States mechanism factors but stays high-level `changelog-20260922-590604e6`

---

08 **Tab group persistence in legacy launch configs** IMPROVED 25

Adds tab group persistence to legacy launch configs.

— Bare one-line description, no mechanism `changelog-20260922-590604e6`

---

09 **Build tool discovery in nested repo directories** IMPROVED 25

Discovers and caches build tools in nested repository directories.

— No detail on which tools or trigger `changelog-20260922-590604e6`

└──▷ ALSO FROM THESE RELEASES

.
.
.

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Warp gains fzf/atuin integration, native shell completions, container registry secret creation, and Windows Materials backdrops.

└──▷ TRY IT

Create a container registry credential secret for a private Docker registry from the CLI.

```
$ oz secret create docker-registry
```

Enable native shell completions as the primary suggestion source instead of the fallback, or disable them entirely, via Warp's terminal input settings.

📍 In Warp, go to Settings › Features › Terminal Input and adjust the native shell completions option.

Apply a Windows Materials backdrop to Warp windows for a modern glass-style appearance.

📍 In Warp on Windows, go to Settings › Appearance › Window › Backdrop and select a Material option.

Adds `oz secret create docker-registry` CLI subcommand to create container registry credential secrets.

Adds native shell completions alongside Warp's built-in completion specs, falling back to native completions when no built-in suggestions exist; configurable in Settings > Features > Terminal Input.

Enables triggering `fzf` (via `ctrl-r` or `ctrl-t`) and `atuin` from within Warp when configured with your shell.

Supports Windows Materials for the backdrop of Warp windows, available in Settings > Appearance > Window > Backdrop.

Adds tab group persistence to legacy launch configs.

+6 more

WAS THIS USEFUL? 👍 👎

Augment Code

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗

RANK

Your engineers have agents. Your organization doesn't.

> Augment Code's IntelliJ and VS Code plugins gained an approval gate before the agent reads sensitive files, proxy support for MCP traffic, and workspace-trust gating for command hooks, alongside a raised minimum JetBrains IDE version and several smaller MCP and chat reliability fixes.

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

01 Configurable IntelliJ sidecar heap limit IMPROVED 93

A new `-Daugment.sidecar.maxOldSpaceSizeMb` IDE system property overrides the sidecar heap ceiling, raised from roughly 1.7 GB to a default of 4096 MB, for loading very large conversations in IntelliJ.

Raise the IntelliJ sidecar heap limit when working with very large conversation histories that exhaust the default 4096 MB ceiling.

env

```
-Daugment.sidecar.maxOldSpaceSizeMb=8192
```

— Exact flag, before/after values, and a runnable example changelog-20260922-24e9a692

02 Proxy-aware MCP connections NEW 83

Remote MCP servers reached over HTTP, SSE, or Streamable HTTP now route through configured proxies: VS Code honors the `HTTP_PROXY` and `HTTPS_PROXY` environment variables for HTTP and SSE servers, and IntelliJ routes MCP traffic reached over HTTP, SSE, or Streamable HTTP through the configured HTTP proxy.

Route MCP server traffic through a corporate proxy on a machine that blocks direct outbound connections — set these before launching VS Code.

```
$ export HTTP_PROXY=http://proxy.corp.example.com:8080
export HTTPS_PROXY=http://proxy.corp.example.com:8080
code .
```

— Names both env vars with a runnable shell example changelog-20260922-24e9a692

03 Rebindable chat submit shortcuts in VS Code NEW 78

Chat submit is now a rebindable keyboard shortcut: `Submit Chat Message` (`Enter`) and `Submit Chat Message Immediately` (`Ctrl+Enter` / `Cmd+Enter`) appear in Keyboard Shortcuts for remapping.

Remap chat submit shortcuts in VS Code so Enter inserts a newline and Ctrl+Enter submits, keeping hands on the keyboard during long agent sessions.

> 📍 In VS Code, open Keyboard Shortcuts (Ctrl+K Ctrl+S), search for `Submit Chat Message`, and rebind `Submit Chat Message` and `Submit Chat Message Immediately` to your preferred keys.

— Names exact shortcuts and where to rebind them `changelog-20260922-24e9a692`

---

04 **Workspace-trust gating for command hooks** BREAKING 78

Workspace-defined Augment hooks in VS Code now run only in trusted workspaces, with workspace-supplied command hooks disabled until workspace trust is granted; in IntelliJ, command hooks in `.augment/settings.json` now run only in projects marked as trusted, while hooks at the user, system, or plugin level continue to run as before.

— Names the config path but no migration steps given `changelog-20260922-24e9a692`

---

05 **Raised minimum JetBrains IDE version** BREAKING 78

The IntelliJ plugin now requires JetBrains IDEs 2025.2 (build 252) or newer; it previously installed on 2024.2 and later. IDEs older than 2025.2 will not be offered this update.

— Exact old and new version thresholds stated clearly `changelog-20260922-24e9a692`

---

06 **Approval gate for sensitive file reads** NEW 73

The agent now asks for approval before reading sensitive files — `.env` files, private keys and certificates, credential- or secret-named files, and paths excluded by `.augmentignore` — in both IntelliJ and VS Code.

— Names exact file types and .augmentignore but no config to tune it

`changelog-20260922-24e9a692`

---

07 **Visible reasons for rejected skills** IMPROVED 65

Skills with invalid `SKILL.md` frontmatter (IntelliJ) or skills in `.augment` that fail to parse (VS Code) are now shown in Settings with the reason they were rejected, instead of disappearing silently.

— Names the file and folder but only a general Settings location

`changelog-20260922-24e9a692`

---

08 **MCP server connection reliability in IntelliJ** IMPROVED

60

MCP servers that announce a changed tool list during a session are now picked up automatically in IntelliJ without requiring a restart, and MCP servers are verified with a real round-trip at startup so a server that completes its handshake but cannot service calls no longer shows as connected.

— Describes mechanism but no user-facing action to take changelog-20260922-24e9a692

THINNER COVERAGE BELOW

09 **Automatic recovery from chat stream interruptions** IMPROVED

50

Chat responses in both IntelliJ and VS Code now automatically retry and recover from mid-stream interruptions, with recovery suppressed after a tool call has already completed.

— Behavioral change described with no configurable surface changelog-20260922-24e9a692

10 **JetBrains 2026.2 IDE support** NEW

40

Adds support for JetBrains 2026.2 IDEs (build 262) in the IntelliJ plugin.

— Just states the supported build with no further detail changelog-20260922-24e9a692

└──▷ ALSO FROM THESE RELEASES

└──▷ BREAKING ON UPGRADE

! The IntelliJ plugin now requires JetBrains IDEs 2025.2 (build 252) or newer; it previously installed on 2024.2 and later. IDEs older than 2025.2 will not be offered this update.

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Augment Code adds sensitive-file approval gates, proxy-aware MCP, rebindable chat shortcuts, and workspace-trust hooks across IntelliJ and VS Code

└──▷ USE IT

Raise the IntelliJ sidecar heap limit when working with very large conversation histories that exhaust the default 4096 MB ceiling.

env

```
-Daugment.sidecar.maxOldSpaceSizeMb=8192
```

Route MCP server traffic through a corporate proxy on a machine that blocks direct outbound connections — set these before launching VS Code.

```
$ export HTTP_PROXY=http://proxy.corp.example.com:8080
export HTTPS_PROXY=http://proxy.corp.example.com:8080
code .
```

Remap chat submit shortcuts in VS Code so Enter inserts a newline and Ctrl+Enter submits, keeping hands on the keyboard during long agent sessions.

📍 In VS Code, open Keyboard Shortcuts (Ctrl+K Ctrl+S), search for `Submit Chat Message`, and rebind `Submit Chat Message` and `Submit Chat Message Immediately` to your preferred keys.

Adds a `-Daugment.sidecar.maxOldSpaceSizeMb` IDE system property to override the sidecar heap ceiling (now 4096 MB, up from ~1.7 GB) for loading very large conversations in IntelliJ.

Remote MCP servers (HTTP and SSE) in VS Code now honor the `HTTP_PROXY` and `HTTPS_PROXY` environment variables.

MCP servers reached over HTTP, SSE, or Streamable HTTP in IntelliJ now route through your configured HTTP proxy.

Chat submit in VS Code is now a rebindable keyboard shortcut — `Submit Chat Message` (`Enter`) and `Submit Chat Message Immediately` (`Ctrl+Enter` / `Cmd+Enter`) appear in Keyboard Shortcuts.

The agent now asks for approval before reading sensitive files — `.env` files, private keys and certificates, credential- or secret-named files, and paths excluded by `.augmentignore` — in both IntelliJ and VS Code.

+7 more

└──▷ BREAKING ON UPGRADE

! The IntelliJ plugin now requires JetBrains IDEs 2025.2 (build 252) or newer; it previously installed on 2024.2 and later. IDEs older than 2025.2 will not be offered this update.

WAS THIS USEFUL? 👍 👎

StackBlitz bolt.new

Feedback ⓘ changelog · Sep 22, 2026

CHANGELOG ↗ RANK

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

> Bolt.new expanded its Forge agent tier with new open-source models and a larger free allowance for Pro users, and introduced a new low-cost Bolt Lite plan.

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

01 **Open-source model tier in Bolt Forge** NEW 60

Bolt Forge now includes an open-source-model tier featuring `GLM 5.3`, `Kimi K3`, and `DeepSeek v4 Pro`, with a larger monthly allowance that is free on Pro plans. Users switch to it via the agent picker in the Bolt editor by selecting `Bolt Forge`, with usage tracked separately from Standard or Max quotas and resetting monthly.

Switch to Forge when you want a larger monthly token budget for exploratory or research workloads, without touching your Standard or Max quota.

> 📍 In the Bolt editor, open the agent picker, select `Bolt Forge`, and accept the terms when prompted. Forge usage is tracked separately and resets monthly.

— Names three models and describes access path, but no allowance figures.

changelog-20260922-bf4c5f06

THINNER COVERAGE BELOW

02 **Bolt Lite standalone plan** NEW 35

A new `Bolt Lite` plan is introduced at $9/month, offering Forge-only access to the core Bolt experience, available via a waitlist.

— States price and access model but no usage steps or scope detail.

changelog-20260922-bf4c5f06

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Bolt Forge adds an open-source-model tier (GLM 5.3, Kimi K3, DeepSeek v4 Pro) with a larger monthly allowance, free on Pro plans.

└──▷ **HOW TO FIND IT**

Switch to Forge when you want a larger monthly token budget for exploratory or research workloads, without touching your Standard or Max quota.

> 📍 In the Bolt editor, open the agent picker, select `Bolt Forge`, and accept the terms when prompted. Forge usage is tracked separately and resets monthly.
>
> Introduces `Bolt Lite`, a new $9/month standalone plan giving Forge-only access to the core Bolt experience via a waitlist.

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗

RANK

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

> Kiro's latest release adds a fullscreen chat mode, enterprise access to Claude Fable 5.1 Preview with a 1M-token context window, a doubled context window for GPT-5.6 models, a redesigned V3 session dashboard, durable agent artifacts, and native ARM64 builds.

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

01 **Claude Fable 5.1 Preview for Enterprise** NEW 90

Adds Claude Fable 5.1 Preview for Kiro Enterprise organizations, featuring a 1M token context window, 6x credit multiplier, and inference limited to `us-east-1`; administrators enable it by adding it from Manage approved list after enabling model access management.

Enable Fable 5.1 for your Enterprise org so users can run repository-scale refactors with its 1M context window.

> 📍 In the Kiro admin console, go to Model Governance › Model access management, enable model access management, then select Manage approved list and add Claude Fable 5.1.

— Full mechanism, limits, and admin enablement steps given changelog-20260922-337914a0

02 **Fullscreen chat mode for TUI sessions** NEW 80

Adds `/fullscreen` command to move the current conversation to a dedicated terminal screen with its own scrollable view, independent of shell history; type `/fullscreen` again to return inline (available in interactive TUI and Lite sessions on V2 and V3). Also adds a Start fullscreen toggle under Display settings to open all future interactive TUI sessions in fullscreen mode by default.

Switch a long-running agentic session to fullscreen mid-conversation to get a clean, scrollable view without shell history clutter.

```
$ /fullscreen
```

Configure Kiro to always open interactive TUI sessions in fullscreen so every new session starts with a dedicated terminal view.

📍 Open Kiro settings, navigate to Display, and turn on `Start fullscreen`.

— Names the exact command, toggle, and settings path changelog-20260922-337914a0

03 **1M-token context for GPT-5.6 models** IMPROVED 73

Expands GPT-5.6 Sol, Terra, and Luna to a 1M token context window (up from 272K) with a two-tier billing model: requests up to 272K tokens bill at the short-context rate (Sol 4.4x, Terra 2.2x, Luna 1.1x), and requests above 272K tokens bill at double that rate.

— Precise limits and pricing but no user action required changelog-20260922-337914a0

04 **Streamlined V3 session dashboard** IMPROVED 70

Introduces a streamlined V3 session dashboard that separates search, filter, sort, and grouping controls from the session list, opens as a flat list ordered by last use, supports sorting by last use, session name, or message count, and remembers your selection; use `Tab` to move between controls and results.

— Describes UI mechanism and keyboard navigation changelog-20260922-337914a0

THINNER COVERAGE BELOW

05 **Durable agent artifacts in Agent Focus** NEW 50

Adds agent artifacts so Kiro can produce durable documents, diagrams, code, and images reviewable in a compact preview or full view beside the chat in Agent Focus, rather than leaving outputs buried in the transcript.

— Explains behavior but no exact command or path changelog-20260922-337914a0

06 **Native ARM64 builds** NEW 40

Adds native ARM64 builds for Windows and Linux, eliminating reliance on x64 emulation.

— Names platforms but no install or verification steps changelog-20260922-337914a0

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Kiro gains Claude Fable 5.1 Preview, fullscreen chat via `/fullscreen`, 1M-token context for GPT-5.6, durable agent artifacts, and native ARM64 builds.

└──▷ TRY IT

Switch a long-running agentic session to fullscreen mid-conversation to get a clean, scrollable view without shell history clutter.

```
$ /fullscreen
```

Enable Fable 5.1 for your Enterprise org so users can run repository-scale refactors with its 1M context window.

> 📍 In the Kiro admin console, go to Model Governance › Model access management, enable model access management, then select `Manage approved list` and add Claude Fable 5.1.

Configure Kiro to always open interactive TUI sessions in fullscreen so every new session starts with a dedicated terminal view.

> 📍 Open Kiro settings, navigate to Display, and turn on `Start fullscreen`.

Adds `/fullscreen` command to move the current conversation to a dedicated terminal screen with its own scrollable view, independent of shell history; type `/fullscreen` again to return inline (available in interactive TUI and Lite sessions on V2 and V3).

Adds a `Start fullscreen` toggle under Display settings to open all future interactive TUI sessions in fullscreen mode by default.

Adds Claude Fable 5.1 Preview for Kiro Enterprise organizations, featuring a 1M token context window, 6x credit multiplier, and inference limited to `us-east-1`; administrators enable it by adding it from `Manage approved list` after enabling model access management.

Expands GPT-5.6 Sol, Terra, and Luna to a 1M token context window (up from 272K) with a two-tier billing model: requests up to 272K tokens bill at the short-context rate (Sol 4.4x, Terra 2.2x, Luna 1.1x), and requests above 272K tokens bill at double that rate.

Introduces a streamlined V3 session dashboard that separates search, filter, sort, and grouping controls from the session list, opens as a flat list ordered by last use, supports sorting by last use, session name, or message count, and remembers your selection; use `Tab` to move between controls and results.

+2 more

WAS THIS USEFUL? 👍 👎

Cognition Devin Desktop

Feedback

changelog · Sep 22, 2026 | CHANGELOG ↗ | RANK

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

> Devin Desktop's biggest window yet: new Windows, macOS, and Android session platforms, a far more extensible Devin CLI (MCP server config, Skills, Hooks, permission rules, new install paths, and a Cloud CLI with SSH and `/handoff`), Azure DevOps support in Devin Review, and a wave of new integrations (PagerDuty, Microsoft Teams, Bitbucket, Linear, Databricks, artifact registries) alongside SAML/OIDC SSO for Enterprise. Devin Desktop itself also gained a remote-session browser preview, a Cascade Migration Wizard, a Plugins section, and a Data Analyst (DANA) agent mode.

└──▷ WHAT SHIPPED · 32 FEATURES most completely described first ? what's the number?

01 **New Devin CLI installation methods** NEW 95

Devin CLI now installs via `brew install --cask devin-cli` on macOS, or on Windows via `irm https://static.devin.ai/cli/setup.ps1 | iex`, which provides native x86_64 and ARM64 binaries usable from PowerShell, Windows Terminal, or Git Bash. For Legacy Windsurf Enterprise and Devin Enterprise plans, the CLI is bundled inside Devin Desktop and installable via the Command Palette (`Cmd+Shift+P` / `Ctrl+Shift+P`) by running `Install Devin CLI`, which adds the `devin` binary to PATH. An `Install Devin CLI in Devin Desktop` admin toggle in Devin CLI team settings enables this bundled install for Enterprise teams.

Install Devin CLI on macOS via Homebrew to get a managed, upgradeable install.

```
$ brew install --cask devin-cli
```

Install Devin CLI on Windows without a GUI installer — run this in PowerShell (not Git Bash or CMD).

```
$ irm https://static.devin.ai/cli/setup.ps1 | iex
```

On a Devin Enterprise plan, install the CLI bundled inside Devin Desktop so the `devin` binary lands on your PATH automatically.

📍 1. Open Devin Desktop. 2. Press Cmd+Shift+P (macOS) or Ctrl+Shift+P (Windows/Linux) to open the Command Palette. 3. Search for and run `Install Devin`

`CLI`. 4. Restart your terminal, then type `devin` in any project directory to start.

— Exact install commands and admin toggle given `product docs`

02 **Devin CLI configuration and extensibility** NEW 93

Devin CLI adds MCP server configuration supporting `stdio`, Streamable HTTP, and SSE transports; `SKILL.md` `-based` Skills — reusable prompt workflows invoked via slash command or by the agent, with their own `allowed-tools` and permissions; Hooks to run shell commands or LLM prompts on lifecycle events, block or rewrite tool calls, and inject context; a `read_config_from` option and a full configuration file covering permissions, proxy, and sandbox settings; permission rules to allow, deny, or prompt for commands, file access, and MCP tools with predictable precedence; and `devin auth` and `mcp` subcommands added to the CLI reference.

Lock down which shell commands Devin CLI can run without prompting by defining allow/deny permission rules in the CLI config file.

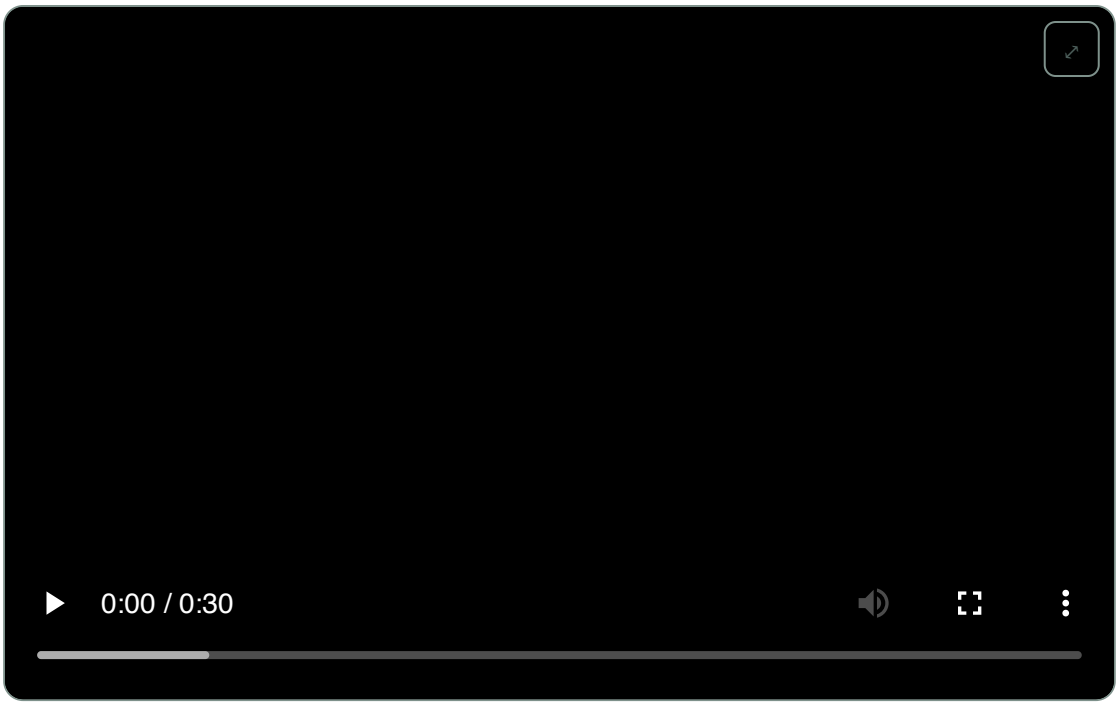
yaml

```yaml
permissions:
  - action: allow
    command: 'npm test'
  - action: deny
    command: 'rm -rf'
  - action: prompt
    command: 'git push'
```

— Names every CLI surface with a working config example `product docs`

03 **Windows, macOS, and Android session platforms** NEW 90

Devin sessions can now run on Windows, macOS, or Android in addition to Linux. Windows desktop sessions support testing Windows-native GUI apps including WPF and WinForms applications, with VM support via `runs-on` and `shell` blueprint fields, Git Bash or PowerShell steps, and Chocolatey installs. macOS sessions support testing Mac and iOS apps via the iOS Simulator, with VM support including Xcode to build, run, and test Apple-platform apps. Android emulator support is available with blueprint setup, `adb` workflows, Computer Use, and session recordings. Platform selection (Linux, Windows, or macOS) is configured via a blueprint, covering Electron apps, IDEs, and platform-native GUIs on each platform; computer use can be enabled or disabled organization-wide via `Settings > Devin > Sessions > Computer use`.



Enable computer use and reach macOS or Windows sessions to test platform-native or iOS apps.

📍 In Devin, go to Settings › Devin › Sessions › Computer use and enable the toggle, then configure a blueprint to select the Windows or macOS platform for your session.

— Names blueprint fields and settings path across all three platforms `product docs`

---

04 **Azure DevOps support in Devin Review** NEW 88

Connect Azure DevOps repositories via the Azure DevOps integration, a Microsoft Entra service principal, or an Azure DevOps Server collection with a personal access token, then enroll them for auto-review from Devin settings, with automated findings posted through the connected service identity. Security vulnerabilities and hardening suggestions are posted as PR comments alongside bug findings in a dedicated Security section of the Analysis sidebar; security scanning runs on every review and cannot be turned off, though posting settings control whether findings post as PR comments, and admins can toggle each finding type (bugs, security findings, investigate flags, informational flags) independently. On mobile, the review opens on the Changes tab, with a switch to the Analysis tab for bugs, flags, and security findings.

Connect an Azure DevOps repository so Devin auto-reviews pull requests and posts findings as PR comments.

📍 In Devin, go to Settings › Devin › Azure DevOps integration, authenticate via a Microsoft Entra service principal or a personal access token for Azure DevOps Server, then enroll the target repository for auto-review.

Review an Azure DevOps pull request directly in Devin Review by pasting its URL.

📍 Paste the full PR URL (for example `https://dev.azure.com/.../pullrequest/`...) into Devin Review to view diffs and analysis.

— Names auth methods, settings paths, and finding-type toggles product docs

05 **Cloud CLI session management via SSH and handoff** NEW

83

A Cloud CLI can create, steer, and resume Devin Cloud sessions from your terminal. `SSH` support opens a shell on a cloud session's VM and forwards ports (for example `devin ssh` and `devin forward 8080`). A `/handoff` command hands off a task from the CLI to a full cloud Devin session and brings that session's resulting pull request back to your local machine.

SSH directly into a running Devin Cloud session's VM to inspect files or run commands, then forward a local port for testing a service running inside the session.

```
$ devin ssh
devin forward 8080
```

Hand off a complex task from the Devin CLI to a full cloud Devin session when local resources aren't enough, then pull the resulting PR back to your machine.

```
$ /handoff
```

— Runnable commands given for SSH, forwarding, and handoff product docs

06 **New third-party integrations** NEW

75

Adds PagerDuty integration to trigger Devin automations from incidents and give sessions access to incidents, services, and on-call schedules; Microsoft Teams integration to start Devin sessions by tagging `@Devin` in a channel, reply in-thread, and control sessions with inline keywords; Bitbucket Cloud and Bitbucket Data Center integration to create pull requests and respond to PR comments via webhook; Linear integration to assign tickets to Devin and convert them into pull requests with playbook labels, `@mentions`, and automation triggers; Databricks integration with service principal, OAuth client secret or OIDC token federation, the Databricks CLI, and Unity Catalog grants; and JFrog Artifactory, AWS CodeArtifact, and Azure Artifacts support by connecting artifact credentials and tools to the environment blueprint.

— Names every integration and its mechanism, but no setup steps product docs

07 **Cascade Migration Wizard** NEW 70

A guided Cascade Migration Wizard helps migrate hooks, workflows, and memories to skills, with per-item selection of what to migrate, a reviewable results screen for each item, and a completion screen at the end.

Migrate existing hooks, workflows, and memories to skills in one guided flow.

> 📍 Open Devin Local, then navigate to Customizations › Cascade Migration Wizard. Select the hooks, workflows, and memories to migrate, review the generated results on the per-item review screen, and confirm on the completion screen.

— Full workflow described with exact navigation path `changelog-20260922-914af932`

---

08 **Computer tab for direct session intervention** NEW 66

A `Computer` tab in the session UI replaces the `Browser` tab when `Computer Use` is enabled for your organization, giving direct view and interaction with Devin's browser and desktop environment. This supports human intervention in Devin's browser sessions for tasks such as completing CAPTCHAs, multi-factor authentication steps, and navigating complex websites.

When Devin stalls on a CAPTCHA or MFA prompt during an automated browser task, switch to the Computer tab to intervene directly and unblock the session.

> 📍 In the Devin session UI, open the `Computer` tab (visible when Computer Use is enabled for your organization) to view and interact with Devin's live desktop and browser environment.

— Names the tab and use cases with a clear starting point `product docs`

---

09 **Automatic post-PR testing and video recordings** NEW 65

Turning on `Pre-approve testing` in Settings > Preferences enables automatic testing without manual approval after a PR is created. Post-PR end-to-end testing also runs with video recordings sent as proof of working changes.

Skip the manual `Run tests` click after every PR — let Devin test automatically as soon as the PR is created.

> 📍 In Devin Desktop, go to Settings › Preferences and enable `Pre-approve testing`.

— Exact settings toggle named with clear navigation `product docs`

---

10 **Plugins section and customization entry points** NEW

65

Devin Local customizations gains a Plugins section where you can review, install, and remove plugins and see the skills, rules, hooks, and MCP servers each one provides. Open customizations entry points were added to the new-tab menu and the Devin Local session context menu, and agent actions menus now show customization options matching the open session.

Review and manage plugins — and the skills, rules, hooks, and MCP servers they expose — without leaving Devin Desktop.

📍 Open Devin Local, then navigate to Customizations › Plugins. Use the Plugins section to install or remove a plugin and inspect which skills, rules, hooks, and MCP servers it provides.

— Groups plugin management with its menu entry points `changelog-20260922-914af932`

11 **Browser preview in remote sessions** NEW 60

Remote agent sessions can now open a browser preview inside Devin Desktop; captured elements and console output from the preview become pending context you can add to the agent composer for your next message.

Open a live browser preview from inside a remote agent session to capture page elements and console output as context for the agent.

📍 In an active remote agent session, go to the agent composer panel and select Open browser preview. Interact with the page; captured elements and console output appear as pending context you can add to your next message.

— UI steps given but no config or API surface named `changelog-20260922-914af932`

THINNER COVERAGE BELOW

12 **Schedule trigger for recurring sessions** NEW 55

A Schedule trigger automates recurring Devin sessions on a daily or weekly cadence, for tasks such as Sentry error triage, dependency updates, and report generation.



Automate weekly dependency update runs so your repo stays current without manual intervention.

📍 In Devin Desktop, open Automations, select `Add a Schedule trigger`, set the cadence to weekly, and enter a prompt such as 'Check for outdated dependencies and open a PR with updates'.

— Names the trigger and gives a concrete setup example `product docs`

---

13 **TestFlight upload from macOS VM** NEW 55

Devin can archive an iOS app on a macOS VM and upload the build to TestFlight via App Store Connect API keys and Devin secrets.

— Names the exact auth mechanism but no step-by-step example `product docs`

---

14 **Data Analyst (DANA) agent mode** NEW 53

A Data Analyst (DANA) agent mode, optimized for querying databases, analyzing data, and creating visualizations, is selectable via Mode > Data in the agent picker.

Switch to the DANA agent when you need Devin to query a database or produce a data visualization instead of writing application code.

📍 1. Open the agent picker. 2. Open the `Mode` submenu. 3. Select `Data` to activate the Data Analyst (DANA) agent. 4. Enter your data task (e.g., 'Query the sales table and chart monthly revenue for Q1').

— Exact menu path given but mechanism not detailed `product docs`

15 **MCP servers surfaced in Devin Local MCP UI** NEW 52

MCP servers from Devin CLI plugins and project-level configurations now appear in the Devin Local MCP UI, including for multi-root workspaces, with an authorize button to restart the OAuth flow for servers that need authorization.

— Describes mechanism but no navigation or example given changelog-20260922-914af932

16 **OIDC workload identity federation** NEW 50

OIDC workload identity federation gives Devin short-lived, keyless access to AWS, GCP, Vault, JFrog, Databricks, and custom services.

— Names supported providers but no setup steps product docs

17 **devin plugins command rename** BREAKING 50

The `devin plugins` command replaces the former singular form; the singular command is no longer accepted.

— Exact CLI command named but no further migration detail changelog-20260922-914af932

18 **SSO setup for Devin Enterprise** NEW 48

SAML SSO setup for Devin Enterprise supports IdP group assertions for group-based access control; OIDC SSO setup supports a generic OpenID Connect identity provider with group claims.

— Names both protocols but no configuration steps product docs

19 **Session list startup speed and automation filtering** IMPROVED 48

Sessions now appear immediately on startup: the agent sidebar and the Sessions board and list views are pre-populated from a per-session cache while the first session list request is still in flight. Automation and scheduled sessions are filtered on the server when the connector supports it, falling back to local filtering otherwise.

— Explains mechanism but nothing for reader to act on changelog-20260922-914af932

20 **Account-scoped /share links** IMPROVED 47

`/share` is now scoped to your account so the generated link works for everyone in the account, it no longer prompts which organization a conversation belongs to, and refusals now explain the server's actual reason for declining.

— Named command with clear before/after but no steps changelog-20260922-914af932

21 **Responding to bots PR comment setting** NEW 45

A Responding to bots PR bot comment setting controls which bot comments Devin responds to: no bots, all bots, or an allowlist.

— Names exact setting and its options `product docs`

22 **Repository indexing from DeepWiki settings** NEW 45

Repository indexing can be triggered from the DeepWiki settings page to power Ask Devin and DeepWiki with up-to-date code understanding.

— Names the settings page as a clear starting point `product docs`

23 **Agents and hooks disabled in Restricted Mode** NEW 42

Agents and hooks are disabled while a workspace is open in Restricted Mode and are automatically re-enabled once you trust the workspace.

— Clear behavior but no config surface to act on `changelog-20260922-914af932`

24 **Devin CLI as ACP agent in Zed editor** NEW 40

Devin CLI can be used as a custom ACP agent inside the Zed editor's Agent Panel, with model selection support.

— Names the integration point but no setup steps `product docs`

25 **Personal Analytics** NEW 39

Personal Analytics lets users view their own ACU consumption and usage analytics from My analytics.

— Names the exact menu location `product docs`

26 **Playbooks** NEW 39

Playbooks are reusable, shareable prompt libraries that can be attached to a session with a macro and browsed from Settings.

— Names macro attachment and Settings location briefly `product docs`

27 **Cascade model picker and Devin Fusion availability** IMPROVED 38

Models marked Devin Local only now appear disabled in Cascade's model picker with a tooltip directing you to Devin Local. Devin Fusion can now be activated even when Claude Fable 5.1 is not available as a lead model.

— Names model picker and Fusion behavior but no steps `changelog-20260922-914af932`

28 **Devin Code Scans** NEW 35

Devin Code Scans find performance, test coverage, dead code, and accessibility issues across repositories.

— Lists scan categories but no run instructions `product docs`

29 **Session Insights** NEW 33

Session Insights analyzes completed sessions, reviews issues and knowledge usage, and surfaces improved prompts for future sessions.

— Describes purpose only, no navigation or mechanism `product docs`

30 **Codemaps icon restored** IMPROVED 32

The Codemaps icon is back in the activity bar in Agent Command Center mode and opens Codemaps in an editor tab.

— Names exact UI location but is a minor restoration `changelog-20260922-914af932`

31 **Plan implementation continues in originating session** IMPROVED 31

Implementing a plan now continues in its originating agent session when available, preserving that session's conversation context instead of starting fresh.

— States behavior change with no steps to invoke `changelog-20260922-914af932`

32 **Devin Review self-serve availability** IMPROVED 18

Devin Review is now available for all self-serve users.

— Bare availability note with no mechanism `changelog-20260922-914af932`

└──▷ ALSO FROM THESE RELEASES

.
.
.

└──▷ BREAKING ON UPGRADE

! The `devin plugins` command replaces the former singular form; the singular command is no longer accepted.

`changelog · Sep 22, 2026` `CHANGELOG ↗` LATEST STABLE

Devin Desktop gains browser preview in remote sessions, a Cascade Migration Wizard, a Plugins section, and per-account `/share` scoping.

└──▷ HOW TO FIND IT

Open a live browser preview from inside a remote agent session to capture page elements and console output as context for the agent.

📍 In an active remote agent session, go to the agent composer panel and select Open browser preview. Interact with the page; captured elements and console output appear as pending context you can add to your next message.

Migrate existing hooks, workflows, and memories to skills in one guided flow.

📍 Open Devin Local, then navigate to Customizations › Cascade Migration Wizard. Select the hooks, workflows, and memories to migrate, review the generated results on

the per-item review screen, and confirm on the completion screen.

Review and manage plugins — and the skills, rules, hooks, and MCP servers they expose — without leaving Devin Desktop.

📍 Open Devin Local, then navigate to Customizations › Plugins. Use the Plugins section to install or remove a plugin and inspect which skills, rules, hooks, and MCP servers it provides.

Remote agent sessions can now open a browser preview inside Devin Desktop, with captured elements and console output available as pending context in the agent composer.

Adds a guided Cascade Migration Wizard that helps you migrate hooks, workflows, and memories, with per-item selection, reviewable results, and a completion screen.

Devin Local customizations now includes a Plugins section where you can review, install, and remove plugins and see the skills, rules, hooks, and MCP servers they provide.

MCP servers from Devin CLI plugins and project-level configurations now appear in the Devin Local MCP UI, including multi-root workspaces, with an authorize button to restart the OAuth flow for servers that need authorization.

Adds `Open customizations` entry points to the new-tab menu and Devin Local session context menu; agent actions menus now show customization options that match the open session.

+9 more

└──▷ BREAKING ON UPGRADE

! The `devin plugins` command replaces the former singular form; the singular command is no longer accepted.

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ desktop-v0.0.33 NOTES ↗ CODE ↗ RANK

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

> Across five releases, Cline shipped git worktree isolation for desktop tasks, made auto-compaction actually trigger reliably on real token usage, ran subagent tool calls concurrently, closed a Windows PATH-hijacking hole, and expanded its model catalog and hook system, alongside a Composio connectors beta.

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

01 Windows working-directory executable search opt-out NEW

95

The CLI and its hub daemon now opt out of working-directory executable search at startup on Windows, preventing a malicious `rg.exe`, `git`, or `powershell` planted in a workspace from running before any approval. `@cline/shared` exposes disableCurrentDirectoryExecutableSearch(), which sets Windows' `NoDefaultCurrentDirectoryInExePath` opt-out and blocks workspace-planted executables like `rg.exe` and `git.exe` from running in place of real ones during file indexing or any spawn.

Protect a Windows host process from workspace PATH-hijacking before any agent spawns subprocesses.

```typescript
import { disableCurrentDirectoryExecutableSearch } from '@cline/shared';

// Call once at process startup, before any spawn
disableCurrentDirectoryExecutableSearch();
```

— Runnable code example plus exact Windows registry mechanism named. sdk/sdk/v0.0.84 cli-v3.0.63

02 Hook system: blocking start, detached reporting, requestId correlation

NEW

90

`beforeRun` hooks can now use the new `blockingRunStartHooks` option to block run start and inject context via the `appendContext` channel on `AgentRunStartResult`, instead of firing detached and discarding results. A new `hooks.detached_runtime` event reports each detached run-start hook once, including a censored sample for hooks still running after a 30-second observation window. `afterModel` hooks now also

receive the `requestId` of the model call that produced the turn, enabling correlation of a model request with the repository state it ran against.

Enable blocking run-start hooks so a `beforeRun` script can inject context into the agent run rather than fire-and-forget.

```yaml
blockingRunStartHooks: true
```

— Names every new hook option, event, and field with a config example. sdk/sdk/v0.0.84

03 **Git worktree isolation for tasks** NEW 85

A Work in switch on the Desktop welcome screen offers a Worktree option that cuts a fresh `cline/<id>` branch off the current one, creates an isolated worktree under `~/.cline/worktrees/`, and runs the task there; deleting the task removes its worktree and branch.

Run an agent task in a fully isolated git worktree so the agent never touches your working tree — useful before a code review or release.

> 📍 In the Desktop app welcome screen, click the Work in switch next to the folder and branch chips, select Worktree, then enter your prompt. Cline cuts a `cline/<id>` branch and creates a worktree under `~/.cline/worktrees/` for the session.

— Full mechanism, exact paths, and a runnable walkthrough. desktop-v0.0.33

04 **Auto-compaction reliability** IMPROVED 85

The 90% context-usage trigger that was never installed in the desktop sidecar is now active. Compaction now fires on the provider's actual token usage — threaded as `previousRequestInputTokens` and evaluated as max(estimate, actual) — instead of a character estimate, so long sessions with dense content (disassembly, image dumps, minified sources) no longer escape the trigger. The default summarizer output budget increases from 4096 to 8192 tokens so a model that reasons by default no longer spends a tight budget thinking and returns no summary at all.

— Detailed mechanism, named internal field and exact token numbers. sdk/sdk/v0.0.84 cli-v3.0.63 desktop-v0.0.33

05 **CLI startup performance improvements** IMPROVED 75

Non-interactive commands such as `cline history --json` no longer load the interactive runtime — on Windows, an isolated history listing drops from 6–7.5 seconds to about 0.85 seconds. Login-shell PATH probing now waits up to 5 seconds per shell instead of 2, so nvm-style profiles on older machines no longer leave runtimes missing

from PATH. Startup no longer waits on a feature-flag network round trip — flags now refresh in the background.

— Concrete before/after timings and an exact command to try. `cli-v3.0.63`

06 **Model catalog and default-model updates** NEW 70

Adds `cline-pass/mimo-v2.6-flash` and `cline-pass/mimo-v2.6-pro` to Cline Pass, and updates default models for 10 providers including Eden AI (Grok 4.7), CoreWeave (DeepSeek V4.1 Flash), and CoralBricks (DeepSeek V4.1 Flash Fast). The catalog expands from 203 to 209 providers and from 6,079 to 6,188 models, with new providers AI21 Labs, ainetcafe, Inco, OCI Generative AI, Tempr, and Vispark, and `kimi.com` / `kimi.ai` split into separate providers.

— Every added provider and model id is named and selectable. `cli-v3.0.63` `sdk/sdk/v0.0.85` `cli-v3.0.64`

07 **Concurrent subagent tool execution** IMPROVED 70

Subagents now run their tool calls concurrently instead of one child finishing before the next starts, via a new `executionMode` on `createTool` for the `spawn_agent` and `subagent_<name>` tools that enables concurrent execution by default — adjacent parallel calls execute together while sequential tools remain ordering boundaries; approvals and hooks still run in order and the parent still waits for every result before its next turn.

— Names the exact API surface and preserves ordering guarantees. `sdk/sdk/v0.0.84` `cli-v3.0.63`

08 **Consistent rules path resolution** IMPROVED 70

`.cline/rules` and global rules are now found consistently across the CLI and the VS Code extension, including in OneDrive-redirected Documents folders. `@cline/shared` now exposes resolveWorkspaceRulesConfigPaths() and resolveGlobalRulesConfigPaths() so hosts stop hardcoding one layout, with the global resolver including OneDrive Known-Folder-Move Documents candidates.

— Names the two resolver functions hosts can call directly. `sdk/sdk/v0.0.84` `cli-v3.0.63`

09 **Output budget scaling for large-output models** IMPROVED 65

The default output budget now scales to max(32,000, floor(maxOutputTokens * 0.3)) for models with large output windows, rather than always capping at a flat 32,000 tokens — a 128,000-token model's budget rises from 32,000 to 38,400.

— Exact formula and worked example, but not user-configurable. `sdk/sdk/v0.0.85` `cli-v3.0.64`

10 **Default model resolution changes for 29 providers** BREAKING

65

The resolved default model changes for 29 providers, including Alibaba CN, Nebius, Neuralwatt, SCNet, Synthetic, TensorX, Vancine, iteracompute, Privatemode AI, SiliconFlow, Volcengine, ZenMux, Zhipu AI, GitHub Copilot, and StepFun among others — sessions on those providers without a pinned model will use a different model after upgrading.

— Lists affected providers and implies a pin-your-model migration step. cli-v3.0.63

11 **Composio connectors beta** NEW

65

Composio connectors, gated behind the `CLINE_COMPOSIO_BETA` account-scoped feature flag, register and execute connector tools through the authenticated Cline API, with the backend owning the Composio project key and scoping connections to the signed-in account.

— Names the gating flag but not usage steps beyond enabling it. sdk/sdk/v0.0.84

12 **Check for Updates in app and tray menus** NEW

60

Adds `Check for Updates…` to the macOS app menu (under About Cline) and to the tray menu on every platform, with a dynamic label reflecting updater state such as `Downloading Update vX.Y.Z…`.

— Clear UI locations named but no deeper mechanism. desktop-v0.0.33

THINNER COVERAGE BELOW

13 **Agent question UX: instant submit and free-text** IMPROVED

55

`agent-ask-question` in `@cline/ui` now submits a single-choice question immediately on selection without a separate Submit step, and gains a free-text input for answers the model did not offer.

— Names the component but is a UI behavior tweak, not directly actionable.

sdk/sdk/v0.0.84 desktop-v0.0.33

14 **Subagent approval delegation change** BREAKING

55

Configured subagents no longer ask for tool-call approval after the parent has already approved delegation. Subagents no longer inherit the parent session's tool approval policies and approval callback, so with manual approvals enabled, a configured child no longer prompts through the parent's approval handler for its own tool calls.

— Describes the behavior change but not a concrete migration step. sdk/sdk/v0.0.84 cli-v3.0.63

15 **Automatic retry on token-exhausted turns** NEW 50

A turn that exhausts its output-token limit without producing a usable tool call is now retried up to three consecutive times, appending a conciseness reminder each time, instead of immediately failing the run.

— Clear mechanism but fully automatic, nothing to configure. cli-v3.0.64

16 **Client-aware recommended-models requests** IMPROVED 45

Requests to the Cline recommended-models endpoint now carry the client type via a client-identity helper in `@cline/shared` and a matching header builder in `@cline/llms`, so recommendations can differ by client.

— Named modules but an internal backend change with no user action. sdk/sdk/v0.0.84

17 **Older Linux x64 machine support** IMPROVED 40

Bundled helpers are now built for pre-AVX2 CPUs so Cline runs on older Linux x64 machines, and Linux SSH remote helpers are compressed to reduce download size.

— Names the constraint but no user-facing action. desktop-v0.0.33

18 **Steer queued message with Enter** NEW 35

Pressing Enter on an empty prompt now steers the first queued message to the front of the queue.

— Simple UI action but no further mechanism given. cli-v3.0.63

19 **CoreWeave provider branding update** IMPROVED 25

CoreWeave now replaces the Weights & Biases branding and docs link for that provider; the stored provider ID and credentials are unchanged.

— Cosmetic rename with no behavioral detail. cli-v3.0.63

╰──▷ BREAKING ON UPGRADE

! The resolved default model changes for 29 providers (including Alibaba CN, Nebius, Neuralwatt, SCNet, Synthetic, TensorX, Vancine, iteracompute, Privatemode AI, SiliconFlow, Volcengine, ZenMux, Zhipu AI, GitHub Copilot, and StepFun among others) — if you use one of those providers without pinning a model, your sessions will use a different model after upgrading.

! Configured subagents no longer inherit the parent session's tool approval policies and approval callback — with manual approvals enabled, a configured child no longer prompts through the parent's approval handler for its own tool calls.

desktop-v0.0.33 NOTES ↗ CODE ↗ LATEST STABLE

Cline Desktop v0.0.33 adds git worktree isolation for tasks, auto-compaction for long sessions, and Check for Updates in app/tray menus.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v0.0.33
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.33
```

└──▷ HOW TO FIND IT

Run an agent task in a fully isolated git worktree so the agent never touches your working tree — useful before a code review or release.

📍 In the Desktop app welcome screen, click the `Work in` switch next to the folder and branch chips, select `Worktree`, then enter your prompt. Cline cuts a `cline/<id>` branch and creates a worktree under `~/.cline/worktrees/` for the session.

Adds a `Work in` switch on the welcome screen with a `Worktree` option that cuts a fresh `cline/<id>` branch off the current one, creates an isolated worktree under `~/.cline/worktrees/`, and runs the task there — deleting the task removes its worktree and branch.

Adds `Check for Updates…` to the app menu on macOS (under About Cline) and the tray menu on every platform, with a dynamic label that reflects updater state (e.g. `Downloading Update vX.Y.Z…`).

Single-choice agent questions now submit immediately on click and accept a typed free-text answer when none of the offered options fit.

Long desktop sessions now auto-compact: the 90% context-usage trigger that was never installed in the sidecar is now active.

Cline now runs on older Linux x64 machines: bundled helpers are built for pre-AVX2 CPUs, and Linux SSH remote helpers are compressed to reduce download size.

cli-v3.0.64 NOTES ↗ CODE ↗ STABLE

Cline CLI v3.0.64 recovers from max-token turns with automatic retries and scales the default output budget for large models.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch cli-v3.0.64 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout cli-v3.0.64
```

Retries a turn that exhausts its output-token limit without producing a usable tool call up to three consecutive times, appending a conciseness reminder each time, instead of immediately failing the run.

Scales the default output budget to 30% of a model's advertised limit (or 32,000 tokens, whichever is larger) for models with large output windows, rather than always capping at a flat 32,000 tokens.

Adds MiMo V2.6 Flash and MiMo V2.6 Pro to Cline Pass, and updates default models for 10 providers including Eden AI (Grok 4.7), CoreWeave (DeepSeek V4.1 Flash), and CoralBricks (DeepSeek V4.1 Flash Fast).

sdk/sdk/v0.0.85 NOTES ↗ CODE ↗ STABLE

Cline: Scales the default output allowance using max(32000, floor(maxOutputTokens * 0.3)) when no request limit or caller default is set and the model advertises an output limit, raising the budget for models above ~106,667 output tokens (e.g. a 128,000-token model goes from 32,000 to 38,400).

└──▷ GET THIS VERSION ▾

```
$ git clone --branch sdk/sdk/v0.0.85
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout sdk/sdk/v0.0.85
```

Adds `cline-pass/mimo-v2.6-flash` and `cline-pass/mimo-v2.6-pro` to the Cline Pass model tier.

cli-v3.0.63 NOTES ↗ CODE ↗ STABLE

Cline CLI v3.0.63: concurrent subagent tool calls, compaction on real token usage, and 6,188 models across 209 providers

└──▷ GET THIS VERSION ▾

```
$ git clone --branch cli-v3.0.63 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout cli-v3.0.63
```

Subagents now run their tool calls concurrently, instead of the first child finishing before the next starts — approvals and hooks still run in order, and the parent still waits for every result before its next turn.

Compaction now fires on your provider's actual token usage instead of a character estimate, preventing long sessions with dense content (disassembly, image dumps, minified sources) from hitting the real context ceiling without ever compacting.

The summarizer gets a larger output budget so a model that reasons by default can no longer spend a tight budget thinking and return no summary at all.

Expands the model catalog from 203 to 209 providers and from 6,079 to 6,188 models — new providers include AI21 Labs, ainetcafe, Inco, OCI Generative AI, Tempr, and Vispark; `kimi.com` and `kimi.ai` split into separate providers.

`.cline/rules` and global rules are now found consistently across the CLI and the VS Code extension, including in OneDrive-redirected Documents folders.

+7 more

└─▷ **BREAKING ON UPGRADE**

! The resolved default model changes for 29 providers (including Alibaba CN, Nebius, Neuralwatt, SCNet, Synthetic, TensorX, Vancine, iteracompute, Privatemode AI, SiliconFlow, Volcengine, ZenMux, Zhipu AI, GitHub Copilot, and StepFun among others) — if you use one of those providers without pinning a model, your sessions will use a different model after upgrading.

sdk/sdk/v0.0.84 NOTES ↗ CODE ↗ STABLE

Cline: Adds disableCurrentDirectoryExecutableSearch() in `@cline/shared` to set Windows' `NoDefaultCurrentDirectoryInExePath` opt-out, blocking workspace-planted executables (`rg.exe` , `git.exe` , etc.) from running in place of real ones during file indexing or any spawn.

└─▷ **GET THIS VERSION** ▾

```
$ git clone --branch sdk/sdk/v0.0.84
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout sdk/sdk/v0.0.84
```

└─▷ **USE IT**

Protect a Windows host process from workspace PATH-hijacking before any agent spawns subprocesses.

```typescript
import { disableCurrentDirectoryExecutableSearch } from
'@cline/shared';

// Call once at process startup, before any spawn
disableCurrentDirectoryExecutableSearch();
```

Enable blocking run-start hooks so a `beforeRun` script can inject context into the agent run rather than fire-and-forget.

```yaml
blockingRunStartHooks: true
```

Adds `blockingRunStartHooks` option so `beforeRun` hooks at the hook-file and subprocess layers can block run start and inject context via the new `appendContext` channel on `AgentRunStartResult`, instead of firing detached and discarding results.

Adds a `hooks.detached_runtime` event that reports each detached run-start hook once, including a censored sample for hooks still running after a 30-second observation window.

Adds resolveWorkspaceRulesConfigPaths() and resolveGlobalRulesConfigPaths() to `@cline/shared`, exposing rules-path resolvers so hosts stop hardcoding one layout — global resolver now includes OneDrive Known-Folder-Move Documents candidates.

Composio connectors, gated behind the `CLINE_COMPOSIO_BETA` account-scoped feature flag, register and execute connector tools through the authenticated Cline API with the backend owning the Composio project key and scoping connections to the signed-in account.

Subagent `spawn_agent` and `subagent_<name>` tools now support a new `executionMode` on `createTool` that enables concurrent execution by default — adjacent parallel calls execute together while sequential tools remain ordering boundaries.

+4 more

└──▷ BREAKING ON UPGRADE

! Configured subagents no longer inherit the parent session's tool approval policies and approval callback — with manual approvals enabled, a configured child no longer prompts through the parent's approval handler for its own tool calls.

WAS THIS USEFUL? 👍 👎

All Hands AI OpenHands

Feedback ⓘ v1.21.0 NOTES ↗

RANK

OpenHands is an open-source software-development agent that writes code, runs commands, and browses the web.

> OpenHands' biggest addition this window is a portable Agent Plugins packaging format for skills and MCP servers, alongside improved automation management in Agent Canvas and the ability to provision Docker execution workspaces from remote Docker servers.

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

01 **Agent Plugins portable packaging format** NEW 94

OpenHands now supports the Agent Plugins portable, client-neutral packaging format via a `plugin.json` manifest at the package root (with `$schema` set to `https://agent-plugins.org/schemas/1.0.0/plugin.schema.json` and a required `name` field). MCP servers load from `mcp.json` (no leading dot) supporting `stdio` and `streamable-http` transports, with deprecated `sse` entries skipped; `${PLUGIN_ROOT}` and `${PLUGIN_DATA}` placeholders are expanded in MCP server `args`, `env`, and `cwd`, and injected as environment variables for `stdio` servers that the package cannot override. Agent Skills load from `skills/<name>/SKILL.md`, or as a single-skill package if only a root-level `SKILL.md` exists. OpenHands-specific extensions (commands, agents, hooks, entry command) live under the `dev.openhands/` namespace, with the entry command declared via `extensions.dev.openhands.entry_command` in `plugin.json` and ignored by other clients. Failures are contained narrowly: an invalid `mcp.json` loads the package with no MCP servers, and an invalid individual entry or skill is skipped while siblings still load. This ships in `openhands-sdk` 1.49.2+, with earlier versions using the legacy `.plugin/plugin.json` layout.

Scaffold a minimal Agent Plugins package with a manifest and an MCP server so it loads in OpenHands and any compatible client without rewrites.

```json
plugin.json (package root):
{
  "$schema": "https://agent-
plugins.org/schemas/1.0.0/plugin.schema.json",
  "name": "code-quality",
  "version": "1.0.0",
  "description": "Linting skills with a code-analysis MCP
server"
}

mcp.json (package root):
{
  "$schema": "https://agent-
plugins.org/schemas/1.0.0/mcp.schema.json",
  "mcpServers": {
    "analyzer": {
      "type": "stdio",
      "command": "./bin/analyzer",
      "args": ["--state", "${PLUGIN_DATA}/cache"],
      "env": { "RULES": "${PLUGIN_ROOT}/rules.toml" }
    }
  }
}
```

Add an OpenHands entry command and hooks to a portable package without affecting other clients that load the same package.

```json
{
  "$schema": "https://agent-
plugins.org/schemas/1.0.0/plugin.schema.json",
  "name": "code-quality",
  "extensions": {
    "dev.openhands": {
      "entry_command": "review"
    }
  }
}
```

— Full manifest schema, config keys, and runnable examples given. product docs

02 **Automation management in Agent Canvas** IMPROVED 65

Automation detail pages now display a disablement-reason banner showing a human-readable reason (such as Manually disabled or Permanent dispatch failure), structured

backend detail, and a timestamp. Agent Canvas also lets you edit an automation's name, prompt, model, timeout, and schedule directly, though cloud backends require manage permissions or that you be the automation's creator.

When an automation stops running unexpectedly, check why and re-enable it from its detail page.

> 📍 1. Open Agent Canvas and navigate to the affected automation's detail page. 2. Read the disablement banner — it shows the reason (`Manually disabled` or `Permanent dispatch failure`), any structured detail from the backend, and the timestamp. 3. If you have manage permissions (or are the automation's creator), use the toggle in the kebab menu or detail view to re-enable it.

Update an automation's schedule or prompt without recreating it from scratch.

> 📍 1. Open Agent Canvas and navigate to the automation's detail page. 2. Select Edit from the kebab menu or detail view. 3. Modify the name, prompt, model, timeout, or schedule fields. 4. Save the changes (cloud backends require manage permissions or that you are the automation's creator).

— Names fields and UI steps but no API/config surface. `product docs`

THINNER COVERAGE BELOW

03 **Remote Docker workspace provisioning** NEW 25

OpenHands can now request Docker execution workspaces from remote Docker servers, enabling distributed agent sandbox provisioning.

— Single vague sentence with no mechanism or config named. `v1.21.0`

v1.21.0 NOTES ↗ LATEST STABLE

OpenHands v1.21.0 adds Docker execution workspace provisioning from remote Docker servers.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.21.0
https://github.com/OpenHands/OpenHands.git
# already have the repo? check out this version:
$ git checkout v1.21.0
```

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.32.16 NOTES ↗ RANK

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

> Letta Code v0.32.16 shipped CLI visibility into agent permissions, subagent launching from App Server clients, and a default switch to the cloud backend on startup.

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

01 **Agent permissions visible in CLI** NEW 30

The CLI now exposes configured agent permissions, including peer names, tracked under `LET-12847`.

— Names a ticket reference but no flag or command shown v0.32.16

02 **Subagent launch from App Server** NEW 23

App Server clients can now launch subagents via the protocol layer.

— States capability but no mechanism or usage path given v0.32.16

03 **Cloud backend default on startup** IMPROVED 23

New sessions now default to connecting to Letta Cloud as the startup backend, rather than requiring explicit configuration.

— Describes the change but no config key or override flag named v0.32.16

v0.32.16 NOTES ↗ LATEST STABLE

Letta Code v0.32.16 defaults to cloud backend, enables subagent launch from App Server, and exposes agent permissions in the CLI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.32.16 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.32.16
```

Exposes configured agent permissions (including peer names) via the CLI with the `LET-12847` feature.

Enables launching subagents from App Server clients via the protocol layer.

> Defaults the startup backend to cloud, so new sessions connect to Letta Cloud out of the box.

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot Feedback ⓘ v10.7.0 NOTES ↗

RANK

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

> Zeroshot v10.7.0 brings the local UI's profile editor and run-history viewer to Cloud, unifying workspace authoring across environments.

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

01 **Shared profile editor and run-history viewer in Cloud** NEW

55

Cloud can now mount the same profile editor and run-history viewer as the local Zeroshot UI, binding drafts, save acknowledgements, and structured streaming errors to their document and authority, enabling shared native authoring and private definition/history exports.

— Describes mechanism but no concrete UI path or config name given. v10.7.0

> v10.7.0 NOTES ↗ LATEST STABLE
>
> Zeroshot v10.7.0 lets Cloud mount the same profile editor and run-history viewer as the local UI, sharing workspace authoring across environments.
>
> └──▷ GET THIS VERSION ▾
>
> ```
> $ git clone --branch v10.7.0 https://github.com/the-open-engine/zeroshot.git
> # already have the repo? check out this version:
> $ git checkout v10.7.0
> ```

WAS THIS USEFUL? 👍 👎

Superset

Feedback ⓘ cli-v1.30.1 NOTES ↗ RANK

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

> Superset's CLI v1.30.1 adds workspace lookup and per-host wake commands, while the desktop app gains a new Mistral Vibe agent, GPT-5.6 models, terminal reliability fixes and UI polish, and the mobile app receives a broad redesign with live ACP agent sessions and deeper Linear integration.

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

01 **Per-host wake commands via CLI** NEW 75

Adds `set-wake` and `wake` CLI subcommands for a configurable per-host wake command, e.g. `superset set-wake <host-id> <wake-command>` followed by `superset wake <host-id>`.

Register a wake command for a host so it can be woken on demand, then trigger it.

```
$ superset set-wake <host-id> <wake-command>
superset wake <host-id>
```

— Named subcommands with runnable example; mechanism not detailed. cli-v1.30.1

02 **`workspaces get` CLI subcommand** NEW 65

Adds a `workspaces get` CLI subcommand to look up a workspace by ID, e.g. `superset workspaces get <workspace-id>`.

Retrieve full details for a specific workspace by its ID from the CLI.

```
$ superset workspaces get <workspace-id>
```

— Exact runnable command given; mechanism otherwise unstated. cli-v1.30.1

03 **Terminal connection reliability and UX in desktop app** IMPROVED 60

Adds a terminal connection status indicator with a diagnosis popover in the desktop app, surfaces why a terminal cannot connect and emits failure telemetry, adds a confirmation dialog before closing a terminal pane with a running process, and adds native-fidelity terminal wheel scrolling via a custom xterm handler and kitty identity.

— Names four terminal changes with some mechanism, no exact steps. cli-v1.30.1

THINNER COVERAGE BELOW

04 **Linear integration: task filtering and Discord triage ingestion** NEW 55

Adds Linear project/cycle filtering and sorting to the tasks view, and auto-ingests Discord support channels into Linear Triage with Claude Sonnet enhancement.

– Names both Linear surfaces but no exact UI path or config. cli-v1.30.1

05 **`cmd+f` search in changes pane** NEW 50

Adds `cmd+f` search to the v2 changes pane in the desktop app.

– Names exact shortcut and location; no further mechanism. cli-v1.30.1

06 **Mistral Vibe agent and GPT-5.6 Codex models** NEW 45

Adds first-class support for the Mistral Vibe coding agent in the desktop app, and adds GPT-5.6 models to the Codex model picker.

– Names the agent and model added but no configuration detail. cli-v1.30.1

07 **Desktop sidebar, settings, and view menu polish** IMPROVED 45

Adds a View menu toggle for the presets bar, adds a tab-drag merge preview plus settings and sidebar UX polish, and replaces the update toast with a sidebar settings-inline updates pill — all in the desktop app.

– Names three UI changes without describing mechanism deeply. cli-v1.30.1

08 **Live ACP agent sessions end-to-end** NEW 45

Adds live ACP agent sessions end-to-end across the host harness, mobile UI, and desktop gate.

– Names the three components spanned but not the mechanism. cli-v1.30.1

09 **Host deletion for workspace owners** NEW 35

Allows workspace owners to delete hosts from the desktop app.

– States the capability but not the UI path or constraints. cli-v1.30.1

10 **Agent failure status surfacing** IMPROVED 35

Surfaces agent failures as a `failed` status in the desktop app and host-service.

– Names status and components but not triggers or UI. cli-v1.30.1

11 **Automations failure badge auto-clear** IMPROVED 35

Clears the automations failure badge when the automations page is opened.

— States exact trigger but nothing more. cli-v1.30.1

12 **Sydney relay region** NEW 30

Adds the Sydney (`syd`) region to the relay fleet, expanding the geographic footprint of relay hosts.

— Bare naming of new region; no further detail. cli-v1.30.1

13 **Local workspace name locked to 'local'** BREAKING 30

Pins the local workspace name to `local` and blocks renaming it.

— States restriction but no rationale or migration note. cli-v1.30.1

cli-v1.30.1 NOTES ↗ STABLE

Superset CLI v1.30.1 adds `workspaces get`, per-host wake commands, Sydney relay region, Mistral Vibe coding agent, and Linear task filtering.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch cli-v1.30.1 https://github.com/superset-sh/superset.git
# already have the repo? check out this version:
$ git checkout cli-v1.30.1
```

└──▷ TRY IT

Retrieve full details for a specific workspace by its ID from the CLI.

```
$ superset workspaces get <workspace-id>
```

Register a wake command for a host so it can be woken on demand, then trigger it.

```
$ superset set-wake <host-id> <wake-command>
superset wake <host-id>
```

Adds `workspaces get` CLI subcommand to look up a workspace by ID.

Adds `set-wake` and `wake` CLI subcommands for a configurable per-host wake command.

> Adds Sydney (`syd`) region to the relay fleet.
>
> Adds live chat sessions over the relay on mobile.
>
> Adds first-class Mistral Vibe coding agent in the desktop app.
>
> +19 more

WAS THIS USEFUL?  

Earendil Works Pi

Feedback ⓘ v0.87.0 NOTES ↗ RANK

AI agent toolkit: unified LLM API, agent loop, TUI, coding agent CLI

> Pi's v0.87.0 release makes SessionManager the canonical source of provider context with append-only context editing, adds new extension lifecycle events (agent_before_settle, agent_settled, context_with_system), and introduces per-model image resize profiles — alongside several breaking changes to turn-ending logic and extension event contracts.

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

01 Canonical session context editing

BREAKING

96

`SessionManager` is now canonical for `AgentSession` provider context: assigning `session.agent.state.messages` no longer replaces future request history. Context can be edited append-only via sessionManager.appendContextEdit(entryId, null) to omit a message from future provider context without altering raw history, usage, or UI history, and compaction can retain nothing via sessionManager.appendCompaction(summary, null, tokensBefore), which stores the compaction's own ID as its kept boundary. A new `ContextEditEntry` is added to the exported `SessionEntry` union (use `replacement: null` for omission); to restore or manage context use SessionManager.inMemory(cwd, { id }, entries), session.navigateTree(), or `session.sessionManager` / session.refreshContext().

Suppress a specific message from future provider requests without touching raw history — useful for redacting sensitive tool output mid-session.

```text
sessionManager.appendContextEdit(entryId, null)
```

— Exact API calls, migration path, and runnable example given. v0.87.0

02 Per-model image resize profiles via inputLimits

NEW

93

Model definitions gain an `inputLimits.images.resize` block (in `models.json`) with `maxWidth`, `maxHeight`, `maxBytes`, and `jpegQuality` fields controlling how images are encoded before entering conversation history, plus a global `images.autoResize` toggle to disable resizing entirely. Catalog fields `inputLimits.maxRequestBytes`, `images.maxPerMessage`, and `images.maxPerRequest` record hard provider limits per model. The selected model's resize profile is applied to `@file` attachments, the `read` tool, and images returned by tools, encoding images once so model switches do not rewrite historical images or invalidate the cached conversation prefix.

Cap image payloads for a custom vision model to stay within a strict provider size limit and reduce bandwidth.

```json
{
  "id": "vision-model",
  "inputLimits": {
    "images": {
      "resize": {
        "maxWidth": 1568,
        "maxHeight": 1568,
        "maxBytes": 524288,
        "jpegQuality": 75
      }
    }
  }
}
```

— Full config schema, defaults, and a runnable example provided. v0.87.0

03 **agent_before_settle and agent_settled lifecycle hooks** BREAKING 88

Extensions gain `agent_before_settle`, which fires at the final actionable boundary of an agent run and lets handlers append `custom`, `custom_message`, `context_edit`, or `compaction` draft entries and request one continuation via `continue: true`; and `agent_settled`, a notification-only hook for status integrations that need to know Pi will not continue running automatically. `ExtensionRunner` exposes emitBoundary(baseEvent, buildContext) for dispatching `agent_before_settle` with chained entry previews, and ExtensionRunner.emit() no longer dispatches actionable turn boundaries (including `turn_end`) — hosts must use `emitBoundary` instead. Runs requested from `agent_settled` handlers are now deferred until all settled handlers finish, eliminating reentrant `agent_start` events during the same notification dispatch, while `TurnEndEvent` gains required `messageEntryId`, `toolResultEntryIds`, `outcome`, and `entries` fields and `AgentBeforeSettleEvent` is added to the exported `ExtensionEvent` union.

— Names exact events, fields, and required host migration. v0.87.0

04 **finishTurn replaces shouldStopAfterTurn** BREAKING 80

The `shouldStopAfterTurn` agent option is removed; migrate to `finishTurn` returning `{ action: 'end' }`. `finishTurn` runs before `turn_end` but applies the decision afterward, and unlike the old option it also receives error and aborted responses — normal-response predicates should return `undefined` for those hard exits.

— Gives exact replacement API and migration nuance. v0.87.0

05 **context_with_system extension event** NEW 76

A new `context_with_system` event fires before each LLM call with the full transcript in `event.messages`, including the leading system message and any mid-conversation prompt/tool patches, giving handlers ownership of the prompt and tool declarations for that request; it runs after `context` handlers and its result is sent verbatim to the provider. The getCurrentSystemMessage() helper from `@earendil-works/pi-ai` lets handlers fold dropped message prefixes into a new head system message when slicing the transcript.

— Names the event, helper function, and exact firing behaviour. v0.87.0

└──▷ BREAKING ON UPGRADE

! Host integrations that construct `TurnEndEvent` values must now provide `messageEntryId`, `toolResultEntryIds`, `outcome`, and `entries` fields.

! `ExtensionEvent` exhaustive switches must also handle `agent_before_settle`.

! ExtensionRunner.emit() no longer dispatches actionable turn boundaries; use emitBoundary(baseEvent, buildContext) instead.

! The `shouldStopAfterTurn` agent option has been removed; migrate to `finishTurn` returning `{ action: 'end' }`. Note: `finishTurn` runs before `turn_end` but applies the decision afterward, and also receives error and aborted responses.

! ExtensionRunner.emit() no longer accepts `turn_end`; host integrations must dispatch actionable boundaries with emitBoundary(baseEvent, buildContext) instead.

! Assigning `session.agent.state.messages` no longer replaces future request history; use SessionManager.inMemory(cwd, { id }, entries), session.navigateTree(), or `session.sessionManager` / session.refreshContext() to manage context.

! `ContextEditEntry` is added to the exported `SessionEntry` union; TypeScript consumers with exhaustive entry switches must handle the `context_edit` case, using `replacement: null` for omission.

! `TurnEndEvent` now has required boundary fields and `AgentBeforeSettleEvent` is added to the exported `ExtensionEvent` union; consumers constructing events or exhaustively switching on `ExtensionEvent` must handle the new shapes.

! The `shouldStopAfterTurn` agent option is removed; replace it with `finishTurn` returning `{ action: "end" }`. `finishTurn` runs before `turn_end`, also receives error and aborted responses, and normal-response predicates should return `undefined` for those hard exits.

! `ContextEditEntry` is added to the exported `SessionEntry` union; TypeScript consumers with exhaustive entry switches must now handle `context_edit`, using `replacement: null` for omission and a content replacement otherwise.

! `SessionManager` is now canonical for `AgentSession` provider context; assigning `session.agent.state.messages` no longer replaces future request history. Restore via SessionManager.inMemory(cwd, { id }, entries), navigate with session.navigateTree(), or append through `session.sessionManager` and call session.refreshContext().

! `TurnEndEvent` is expanded with required boundary fields and `AgentBeforeSettleEvent` is added to the exported `ExtensionEvent` union; consumers constructing events or exhaustively switching on `ExtensionEvent` must handle the new shapes. ExtensionRunner.emit() no longer accepts `turn_end`; host integrations must dispatch actionable boundaries with emitBoundary(baseEvent, buildContext).

! Runs requested from `agent_settled` handlers are now deferred until all settled handlers finish; handlers still observe `ctx.isIdle() === true` but no longer see a reentrant `agent_start` during the same notification dispatch.

v0.87.0 NOTES ↗ LATEST STABLE

Pi v0.87.0 adds canonical session context editing, `context_with_system` extension events, and per-model image resize profiles via `inputLimits.images.resize`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.87.0 https://github.com/earendil-works/pi.git
# already have the repo? check out this version:
$ git checkout v0.87.0
```

└──▷ USE IT

Suppress a specific message from future provider requests without touching raw history — useful for redacting sensitive tool output mid-session.

```text
sessionManager.appendContextEdit(entryId, null)
```

Adds `context_with_system` extension event, which runs after `context` handlers on the full transcript including system messages and sends its result verbatim to the provider.

Adds per-model image resize profiles through `inputLimits.images.resize` in `models.json`, applied to file attachments, image reads, and tool-result images.

Adds append-only model-context edits via sessionManager.appendContextEdit(entryId, null) to omit messages from future provider context without altering raw history, usage, or UI history.

Adds actionable `turn_end` and `agent_before_settle` extension boundaries, allowing extensions to persist structural entries and ensure a next provider request.

Adds retain-none compaction input via sessionManager.appendCompaction(summary, null, tokensBefore), storing the compaction's own ID as its kept boundary.

+1 more

└──▷ BREAKING ON UPGRADE

- ! The `shouldStopAfterTurn` agent option is removed; replace it with `finishTurn` returning `{ action: "end" }`. `finishTurn` runs before `turn_end`, also receives error and aborted responses, and normal-response predicates should return `undefined` for those hard exits.
- ! `ContextEditEntry` is added to the exported `SessionEntry` union; TypeScript consumers with exhaustive entry switches must now handle `context_edit`, using `replacement: null` for omission and a content replacement otherwise.
- ! `SessionManager` is now canonical for `AgentSession` provider context; assigning `session.agent.state.messages` no longer replaces future request history. Restore via SessionManager.inMemory(cwd, { id }, entries), navigate with session.navigateTree(), or append through `session.sessionManager` and call session.refreshContext().
- ! `TurnEndEvent` is expanded with required boundary fields and `AgentBeforeSettleEvent` is added to the exported `ExtensionEvent` union; consumers constructing events or exhaustively switching on `ExtensionEvent` must handle the new shapes. ExtensionRunner.emit() no longer accepts `turn_end`; host integrations must dispatch actionable boundaries with emitBoundary(baseEvent, buildContext).
- ! Runs requested from `agent_settled` handlers are now deferred until all settled handlers finish; handlers still observe `ctx.isIdle() === true` but no longer see a reentrant `agent_start` during the same notification dispatch.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ desktop-v0.24.3 NOTES ↗

RANK

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

> Qwen Code's desktop release adds a Linux kernel sandbox, a Playwright browser SDK, DingTalk output, and a wide set of workflow, Goal, and Web Shell controls, while the TypeScript SDK gains multimodal input, session resume/export, an LSP, a Zed extension, and a breaking settings-key rename from `disable*` to `enable*`.

└──▷ **WHAT SHIPPED · 21 FEATURES** most completely described first ? what's the number?

01 Workflow tool execution controls, budgeting, and UI safeguards

NEW 87

Adds `tools.workflowNameOnly`, a name-only lock restricting the Workflow tool to named workflows, plus agent() narrowing so a subagent can be restricted to an explicit tool allowlist. Adds a `+Nk` directive to share a per-turn token budget across workflow steps, budget-based ACP child admission with idle-child reclamation when admission is full, and the ability to run saved workflows by name while pinning workflow grants to script content. Adds `sourceRef` and caller-supplied script support when starting workflow runs via the serve API, summary projection support for prompt and replay daemon APIs, extension workflow invocation by the model when a workflow declares `whenToUse`, correlation of agent dispatches with nested workflow calls, and a workflow size guideline, structure preview, and large-run warning in the workflows UI.

Lock the Workflow tool to named workflows via the `tools.workflowNameOnly` setting so the model cannot run ad-hoc scripts.

```json
{
  "tools": {
    "workflowNameOnly": true
  }
}
```

— Names every workflow flag and API surface added this window desktop-v0.24.3

02 Session resume and continuation support NEW 70

Adds `resume` and `continue` options plus extended `authType` support to the TypeScript SDK, and support for resuming sessions with a session ID in query mode.

— Names exact options and modes for resuming sessions sdk-typescript-v0.1.14

03 **Session export command** NEW — 70

Adds an `export` command for session history in markdown and HTML formats.

— Exact command and output formats given `sdk-typescript-v0.1.14`

04 **SDK configuration additions** NEW — 68

Adds a `settings.env` field for environment variable configuration directly in settings, `contextWindowSize` configuration support, and `extra_body` support for OpenAI-compatible providers.

— Names three exact config keys added `sdk-typescript-v0.1.14`

05 **DingTalk output and shared output modes** NEW — 65

Adds shared output modes with DingTalk support, including sender identity in direct messages, invoked via `qwen channel start`.

— Names the exact command to start the channel `desktop-v0.24.3`

06 **Settings keys renamed from disable* to enable*** BREAKING — 65

Settings keys previously named with a `disable*` prefix are renamed to `enable*` positive naming (e.g. `disableX` becomes `enableX`); existing configs using the old names will break.

— Gives exact rename pattern and migration impact `sdk-typescript-v0.1.14`

07 **Model and provider configuration additions** NEW — 60

Adds support for external reasoning profiles and defaults in core model configuration, the ability to select the OpenAI wire API per model, and configurable Mem0 providers under `external-context` settings.

— Names config surface but no mechanism detail given `desktop-v0.24.3`

08 **Web Shell UI enhancements** NEW — 60

Adds footnote previews and per-turn source citations, file preview from expanded tool details, remote daemon connection, git remote management from the workspace branch picker, oversized-paste folding into an attachment card, and host settings item exclusions support in the Web Shell.

— Names each UI addition but no exact path shown `desktop-v0.24.3`

09 **FORK_MODE for ProcessTransport** NEW — 60

Adds `FORK_MODE` support to `ProcessTransport` for Electron IPC integration.

— Names exact mode and transport but no usage steps `sdk-typescript-v0.1.14`

THINNER COVERAGE BELOW

10 **Multimodal input pipeline** NEW 57

Integrates a multimodal media pipeline (`omni`) into main and adds multi-modal input support (image, PDF, audio) across all content generators.

– Names supported media types but no usage path desktop-v0.24.3 sdk-typescript-v0.1.14

11 **Coding Plan authentication mode** NEW 57

Adds a Coding Plan authentication mode with a unified `AuthDialog` and configuration management.

– Names the auth dialog surface but no setup steps sdk-typescript-v0.1.14

12 **Goal lifecycle telemetry, budgeting, and billing** NEW 54

Adds Goal lifecycle telemetry events and metrics, turn and active-time budget display in the Goal view, Goal state records in session exports, and foreground subagent and verifier token usage billing to Goal.

– Lists surfaces but no navigation or command given desktop-v0.24.3

13 **Hook system improvements** NEW 53

Adds hook invocation identification in progress events and message bus announcements, plus hook registry reload when `/hooks` is opened in the CLI.

– Names the /hooks command as an entry point desktop-v0.24.3

14 **Agent Skills maturity and management** IMPROVED 50

Promotes Agent Skills from experimental to stable, adds symlink support for the skill manager, and adds local enumeration of installed extension skills from the daemon.

– Names three skill-related changes without mechanism desktop-v0.24.3 sdk-typescript-v0.1.14

15 **Playwright-based Browser SDK** NEW 45

Adds a Playwright-based Browser SDK (`browser-use`) for programmatic browser automation.

– Names the module but no usage detail desktop-v0.24.3

16 **Concurrent batch runner** NEW 45

Adds a concurrent runner for batch CLI execution.

– Describes purpose but no command shown sdk-typescript-v0.1.14

17 **MCP tool progress updates** NEW 43

Adds MCP tool progress update support in TUI and SDK mode.

– Names surfaces affected without further mechanism sdk-typescript-v0.1.14

18 **Zed extension for agent server** NEW 40

Adds a Zed extension for the Qwen Code agent server.

– Names the integration target but no install steps sdk-typescript-v0.1.14

19 **Experimental LSP support** NEW 37

Adds experimental LSP support for code intelligence.

– Bare feature name with no configuration detail sdk-typescript-v0.1.14

20 **macOS paste and submit fixes** IMPROVED 30

Adds a large paste placeholder and fixes enter-submit behavior on macOS.

– Minor fix described in one line sdk-typescript-v0.1.14

21 **Cross-session messaging enabled by default** IMPROVED 25

Cross-session messaging is now enabled by default.

– States the default change with no further detail desktop-v0.24.3

└──▷ BREAKING ON UPGRADE

! Settings keys previously named with a `disable*` prefix are renamed to `enable*` positive naming (e.g. `disableX` becomes `enableX`); existing configs using the old names will break.

desktop-v0.24.3 NOTES ↗ LATEST STABLE

Qwen Code desktop-v0.24.3 adds a Linux kernel sandbox, Playwright browser SDK, DingTalk output, multimodal pipeline, and per-model OpenAI wire selection.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v0.24.3
https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout desktop-v0.24.3
```

└──▷ USE IT

Lock the Workflow tool to named workflows via the `tools.workflowNameOnly` setting so the model cannot run ad-hoc scripts.

```json
{
  "tools": {
    "workflowNameOnly": true
  }
}
```

Adds `bwrap` kernel sandbox backend for Linux, isolating CLI tool execution at the OS level.

Adds a Playwright-based Browser SDK (`browser-use`) for programmatic browser automation.

Adds shared output modes with DingTalk support, including sender identity in direct messages, via `qwen channel start` .

Adds support for external reasoning profiles and defaults in core model configuration.

Adds ability to select the OpenAI wire API per model.

+26 more

sdk-typescript-v0.1.14 NOTES ↗ STABLE

Qwen Code SDK v0.1.14 adds multi-modal input, session resume, `settings.env` , export command, concurrent batch runner, LSP, and Zed integration.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch sdk-typescript-v0.1.14
https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout sdk-typescript-v0.1.14
```

Adds `settings.env` field for environment variable configuration directly in settings.

Adds `resume` and `continue` options plus extended `authType` support to the TypeScript SDK.

Adds `export` command for session history in markdown and HTML formats.

Adds `extra_body` support for OpenAI-compatible providers.

Adds `contextWindowSize` configuration support.

+11 more

└──▷ BREAKING ON UPGRADE

> ! Settings keys previously named with a `disable*` prefix are renamed to `enable*` positive naming (e.g. `disableX` becomes `enableX`); existing configs using the old names will break.

WAS THIS USEFUL? 👍 👎

GitHub Copilot CLI

Feedback ⓘ v1.0.87 NOTES ↗ RANK

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

> Copilot CLI v1.0.87 adds configurable worktree paths, command-line visibility into the built-in GitHub MCP server, and several agent and sandbox refinements including pending-prompt recall and cross-platform proxy support.

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

01 **Configurable worktree path template** NEW 90

A new `worktreePathTemplate` setting controls where `/worktree`, `/move`, `/new`, and `--worktree` create worktrees, supporting `{repoPath}`, `{repo}`, `{branch}`, and `{branchSlug}` placeholders instead of the default `<repo>.worktrees/` layout.

Keep all worktrees for every repo organized under a single directory tree, named by repo and branch, instead of the default `<repo>.worktrees/` layout.

yaml

```yaml
worktreePathTemplate: ~/src/worktrees/{repo}/{branch}
```

— Named config key, placeholders, and a runnable example given v1.0.87

02 **CLI visibility for built-in GitHub MCP server** IMPROVED 83

`copilot mcp list` and `copilot mcp get` now report the built-in `github-mcp-server` when signed in, whereas previously it only appeared in the interactive `/mcp` view.

Verify the built-in GitHub MCP server is registered and active from the command line without entering an interactive session.

```
$ copilot mcp get github-mcp-server
```

— Exact subcommands and a runnable verification command v1.0.87

03 **Configurable MCP slow-connection threshold** NEW 83

Adds a `slowConnectionThresholdMs` config key to set per-server MCP slow-connection warning thresholds, e.g. to silence warnings for fast local servers while keeping defaults for remote ones.

Silence noisy `slow connection` warnings for a fast local MCP server while keeping the default threshold for remote ones.

```yaml
slowConnectionThresholdMs: 5000
```

— Named config key with a concrete usage example v1.0.87

THINNER COVERAGE BELOW

04 **Pending-prompt recall for steering prompts** IMPROVED 55

Consecutive steering prompts sent in the same mode now combine into a single pending message; pressing Up in an empty chat input recalls it for editing, including any pasted text and attachments.

— Describes mechanism but no command or config surface v1.0.87

05 **Strict marketplace allowlist enforcement** BREAKING 55

Empty `strictKnownMarketplaces` allowlists now hide and block built-in plugin marketplaces, changing prior behavior where an empty list had no restrictive effect.

— Named config key with before/after but no example v1.0.87

06 **Auto routing tier startup defaults and org policy** NEW 40

Adds user and managed startup defaults for the Auto routing tier, including strict and user-overridable organization policy controls.

— Names the feature area but no config key or command given v1.0.87

07 **Rubber-duck agent for all model families** IMPROVED 35

The rubber-duck agent is now enabled for every model family and for low-cost-tier session models.

— States scope but no mechanism or usage path v1.0.87

08 **Cross-platform sandbox proxy support** IMPROVED 30

Sandbox proxies now work on Windows, and a proxy configured with a username and password works on every platform.

— Bare statement of what now works, no config detail v1.0.87

09 **Live elapsed time for execution subagents** IMPROVED 25

Timeline entries now show live elapsed time for execution subagents while they run.

— Simple UI change with no further detail v1.0.87

v1.0.87 NOTES ↗ LATEST STABLE

Copilot CLI v1.0.87 adds `worktreePathTemplate`, pending-prompt recall, and `copilot mcp list/get` visibility for the built-in GitHub MCP server.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.0.87 https://github.com/github/copilot-cli.git
# already have the repo? check out this version:
$ git checkout v1.0.87
```

└──▷ USE IT

Keep all worktrees for every repo organized under a single directory tree, named by repo and branch, instead of the default `<repo>.worktrees/` layout.

```yaml
worktreePathTemplate: ~/src/worktrees/{repo}/{branch}
```

Silence noisy slow connection warnings for a fast local MCP server while keeping the default threshold for remote ones.

```yaml
slowConnectionThresholdMs: 5000
```

Verify the built-in GitHub MCP server is registered and active from the command line without entering an interactive session.

```
$ copilot mcp get github-mcp-server
```

Adds `worktreePathTemplate` setting to control where `/worktree`, `/move`, `/new`, and `--worktree` create worktrees; supports `{repoPath}`, `{repo}`, `{branch}`, and `{branchSlug}` placeholders.

`copilot mcp list` and `copilot mcp get` now report the built-in `github-mcp-server` when signed in, instead of showing it only in the interactive `/mcp` view.

Adds `slowConnectionThresholdMs` config key to set per-server MCP slow-connection warning thresholds.

> Consecutive steering prompts in the same mode now combine into one pending message; press Up in an empty chat input to recall it for editing, including pasted text and attachments.
>
> Enables the rubber-duck agent for every model family and for low-cost-tier session models.
>
> +4 more

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ

One Runner, Many Worktrees (2026-09-22) BLOG ↗ RANK

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

> Amp's biggest window yet: Orbs (remote, isolated execution environments) shipped with a Desktop tab, Portals for authenticated service exposure, Automations for scheduled threads, and worktree-aware runners, alongside native macOS/iOS apps, a Puck personal assistant, in-thread Space voice/video calls, and a sweeping CLI expansion covering a full settings system, streaming JSON output, execute/orb-execute modes, shell completions, and hosted plugin/skill repositories.

└──▷ **WHAT SHIPPED · 20 FEATURES** most completely described first ? what's the number?

01 Runner directories, secrets, and worktree creation NEW

96

Runners gained directory management, secrets injection, and worktree creation. `--no-tui` starts a headless runner that executes remotely created threads; `--runner-id <id>` assigns a stable identifier, targetable via `--executor runner:<id>` on `amp -x`. `--dir <path>` serves additional directories, `--discover-dirs` / `--discover-dirs=<path>` auto-serves every Git checkout up to two levels deep with live filesystem watching (macOS/Linux), `--discover-depth <levels>` (1–10) scans deeper, `--discover-exclude <pattern>` skips `.gitignore` `-style` matches, and `--no-serve-cwd` stops auto-serving the working directory; `amp runner dirs add`, `amp runner dirs list`, and `amp runner dirs remove` change served directories live. `--amp-env` (and the `amp.runner.env.enabled` config key) injects ampcode.com-managed Secrets & Env Vars into runner threads with precedence personal > project > workspace > runner process; `amp.runner.autoUpdate.enabled` lets a runner auto-install releases (checked hourly, restarted at most once per 12 hours). The runner directory picker on ampcode.com can create new directories and Amp-hosted projects on the remote machine, and New Worktree runs `git worktree add` to create a sibling checkout (e.g. `~/code/amp-fix-flaky-login-test` on branch `fix-flaky-login-test`) and starts the thread there; an Archive and Remove Worktree thread action then archives the thread, runs `git worktree remove`, and deletes the branch, and Puck can create worktrees programmatically by passing a worktree name to `create_thread`.

⋮

Start a persistent headless runner on a remote build server, exposing multiple Git checkouts and its terminal, so teammates can launch threads from ampcode.com without SSH access.

```
$ amp --no-tui --runner-id grandmas-garage-server --remote-control-terminal --discover-dirs=~/code --discover-depth 3 --discover-exclude 'clients/*/vendor'
```

Dispatch a task to a specific named runner from CI or another terminal, keeping local and remote execution separate.

```
$ amp -x "Run the tests and fix any failures" --executor runner:grandmas-garage-server
```

Enable ampcode.com-managed secrets on a long-running runner so you can rotate API keys from the web without restarting the process.

```yaml
amp.runner.env.enabled: true
amp.runner.autoUpdate.enabled: true
```

Start a runner that automatically injects ampcode.com secrets into every new thread — useful when agents need API keys or tokens without hardcoding them.

```
$ amp --no-tui --runner-id mac-mini --discover-dirs --amp-env
```

— Full flag/config precedence and worktree lifecycle with runnable commands.

One Runner, Many Worktrees (2026-09-22)

---

02 **Orbs: remote execution environment** NEW 94

Orbs are remote, isolated machines where Amp agents run tasks independently with their own filesystem, services, and terminal, without consuming local compute. `-ox` (shorthand for `--orb-execute -x`) runs an execute-mode prompt in a cloud orb, printing the thread URL and returning immediately; `--attach <path>` uploads local files into the prompt (repeatable), `--executor orb` opens the interactive TUI on a new orb thread awaiting the first prompt, and `--executor local` / `--executor runner:<runner-id>` explicitly choose execution location. `--project <namespace/name|owner/repo|url>` disambiguates which project an orb starts from, and `--orb-size <size>` (e.g. `a1.large`) picks a non-default machine size. `--stream-json` works with `-ox` to stream orb thread events in the same format as local `amp -x --stream-json`, and `--no-archive-after-execute` stops `-ox --stream-json` from archiving the thread when the turn ends; `-ox` can be combined with `amp threads continue <thread-id>` to message an existing orb thread from CI or scripts, authenticated via the `AMP_API_KEY` environment variable (an `sgamp_` `-prefixed`

token), and supports piped stdin like local `-x`. Each orb exposes a shared `tmux` session in its Terminal pane, accepts file uploads up to 512 MB via the Files or Terminal pane, and is subject to a per-user spawn burst limit of 20 metered orbs before queuing at one every five minutes with an estimated wait shown in the thread; `amp sync <thread>` pulls an orb agent's in-progress changes into the local checkout.

Kick off a long-running investigation from CI without blocking the pipeline — the agent runs on Amp's servers and reports back asynchronously.

```
$ git diff main | amp -ox "Investigate any DB perf risks in this diff, report back to me on Slack" --project my-org/my-repo
```

Stream orb agent output as newline-delimited JSON in a script so you can parse turn events programmatically, using a long-lived access token in CI.

```
$ export AMP_API_KEY=sgamp_your-token-here
amp -ox "Summarize the open TODOs in this repo" --stream-json --no-archive-after-execute
```

Wake an existing orb thread from a deploy script to trigger post-deploy verification without creating a new thread.

```
$ amp threads continue T-… -ox "The deploy finished, verify the fix in production"
```

Kick off a long-running investigation in an orb from your terminal so it keeps running after you close your laptop.

```
$ amp -ox "Investigate why the latest CI run on main failed"
```

Run a compute-heavy task (like a performance test matrix) in a larger orb instance.

```
$ amp -ox "Run the performance test matrix" --orb-size a1.large
```

Pull an orb agent's in-progress changes into your local checkout to review or test them without waiting for the agent to finish.

```
$ amp sync <thread>
```

— Every orb flag and limit named with runnable CI examples. product docs

03 Portals: authenticated service exposure for orbs NEW

94

`.amp/services.yaml` declares long-running services with `command`, `cwd`, `port`, `env`, `health`, `portal`, `portals`, `review`, `agent`, and `platforms` fields. `amp orb services ensure` (with `--json`) starts all declared services and prints portal URLs; `amp orb service start` (`--command`, `--portal`, `--title`, `--description`, `--cwd`, `--port`) creates one-off thread-scoped services; `amp orb portal <port>` (`--title`, `--description`, `--hostname`, `--take-over`) creates or prints a portal URL for a running server; and `amp orb service list`, `status <name>`, `logs <name>`, `restart <name>`, and `stop <name>` inspect and control services. Portal links are written to `.amp/portals/*.json`, `.agents/setup` and `.agents/resume` hooks prepare and repair orb environments, `${services.<name>.publicURL}` lets one service reference another's portal URL, and `$AMP_USER_EMAIL` can appear (URL-encoded in URLs) in `url` / `title` / `description`. `agent: observe` lets the agent read browser logs/errors and `agent: control` allows page inspection/control from the Portal tab; `review: false` or the `x-amp-review-widget: off` response header disables the review widget, `x-amp-portal-agent: off` disables agent access per response, and `platforms: [linux]` / `[darwin]` restricts a service to one OS. Custom hostnames work via `amp orb portal <port> --hostname <hostname>` and workspace Domains defaults, with `--take-over` moving a hostname from an archived thread's portal, and portal link folders group related links under `portals` using `folder` / `links`.

Declare a web service with a health check and portal so every orb starts the dev server consistently and exposes an authenticated URL.

yaml

```yaml
# .amp/services.yaml
services:
  web:
    command: pnpm dev -- --host 0.0.0.0 --port "$PORT"
    health: /healthz
    env:
      API_MODE: development
    portal:
      url: /
      title: App
      description: Use the seeded development account.
    portals:
      - url: /admin
        title: Admin
        description: Open the administration tools.
```

Start all declared services and get structured portal URLs to feed into a CI or scripting pipeline.

```
$ amp orb services ensure --json
```

Spin up a one-off preview server in the current thread and get its authenticated portal URL without modifying the shared services config.

```
$ amp orb service start preview \
  --command 'npm run preview -- --host 0.0.0.0 --port "$PORT"' \
  --portal \
  --title 'Preview' \
  --description 'Use the seeded development account.'
```

— Every config field, subcommand, and header enumerated. `product docs`

04 **CLI settings system** NEW 88

A full CLI settings system spans user, workspace, and enterprise-managed configuration. `--settings-file <path>` points to a custom user settings file, and `amp config edit` (with `--workspace`) opens settings in `$EDITOR`. Keys in `settings.json` / `settings.jsonc` include `amp.fuzzy.alwaysIncludePaths` (glob patterns always included in fuzzy search), `amp.showCosts` (thread cost display), `amp.git.commit.ampThread.enabled` and `amp.git.commit.coauthor.enabled` (control the Amp-Thread-ID and `Co-authored-by: Amp` commit trailers, also toggleable via `AMP_DISABLE_AMP_THREAD_TRAILER` / `AMP_DISABLE_AMP_COAUTHOR_TRAILER`), `amp.mcpServers` and `amp.mcpPermissions` (allow/reject MCP servers by `command` / `args` or `url`, first match wins), `amp.defaultVisibility` (`private` / `workspace` / `group` per repo origin), `amp.remoteThreadCreation.enabled` (lets ampcode.com open new threads in the local TUI), `amp.runner.autoUpdate.enabled` and `amp.runner.env.enabled` (runner update and secrets injection, equivalent to `--amp-env`), `amp.skills.disableClaudeCodeSkills`, `amp.skills.disableGlobalAgentsSkills`, and `amp.skills.path` (skill directory control), `amp.tools.disable` (disable tools by name or `builtin:toolname` / `*` glob), `amp.keymap` (custom keybindings, user overrides workspace), `amp.notifications.enabled` (sound or terminal bell over SSH / `AMP_FORCE_BEL`), `amp.terminal.copyOnSelect`, `amp.terminal.detailsExpandedByDefault` (toggle live with Alt+T), `amp.thread.autoArchiveOnQuit`, and `amp.updates.mode` (`auto` / `warn` / `disabled`, or `AMP_SKIP_UPDATE_CHECK=1`). Enterprise managed settings apply via a workspace JSON document (Workspace Settings > Advanced > Managed Settings) or local files at `/Library/Application`

`Support/ampcode/managed-settings.json` (macOS), `/etc/ampcode/managed-settings.json` (Linux), or `%ProgramData%\ampcode\managed-settings.json` (Windows), including `amp.admin.compatibilityDate` for migration control; `settings: show managed` lists managed keys and merge behavior, and a JSON Schema is published at `https://ampcode.com/cli-settings.schema.json`. Settings are read from `~/.config/amp/settings.json` / `.jsonc` (or Windows equivalent) and workspace-level `.amp/settings.json` / `.jsonc` searched upward to the repository root.

Open your user settings file directly in your editor to review or change any setting.

```
$ amp config edit
```

— Every settings key, default, and file path enumerated. product docs

---

05 **Orb lifecycle hooks and snapshot management** NEW 88

`amp projects snapshots list <project>` and `amp projects snapshots delete <project> --resource <size>` inspect and purge cached orb snapshots (omit `--resource` to delete all). `amp orb id-token --audience <url> --subject-scope project|workspace` mints OIDC tokens inside `.agents/setup` or pre-setup scripts for secret-free authentication, and `amp projects update <namespace/name> --pre-setup-script-file <path>` pushes a pre-setup script to project settings without a repo commit; Docker can run as a supervised service via `amp orb service start docker-daemon --command 'sudo dockerd'`. The committed `.agents/setup` hook runs during orb provisioning (its output cached as a reusable snapshot for up to 72 hours) and `.agents/resume` runs after each activation and wake using per-thread workload identity credentials; project settings add a Pre-clone script (runs from `/home/user/workspace` before cloning, failure stops setup) and a Pre-setup script (runs immediately before `.agents/setup`). Setup and resume logs are written to `/home/user/.cache/amp/logs/setup.log` and `resume.log`, `GIT_CONFIG_COUNT` / `GIT_CONFIG_KEY_0` / `GIT_CONFIG_VALUE_0` env vars configure HTTPS credentials for private non-GitHub repos, and `.amp/services.yaml` is offered as an alternative to starting processes from `.agents/setup`.

Force-refresh a stale orb snapshot after changing your setup script, targeting a specific orb size.

```
$ amp projects snapshots list myorg/myrepo
amp projects snapshots delete myorg/myrepo --resource a1.small
```

Mint a project-scoped OIDC token inside `.agents/setup` to authenticate to a secrets manager or artifact registry without storing long-lived credentials.

```
$ amp orb id-token \
  --audience https://artifactory.example.com \
  --subject-scope project
```

Push a pre-setup script to project settings so orbs run it before `.agents/setup` without requiring a repository commit.

```
$ amp projects update myorg/myrepo --pre-setup-script-file
./pre-setup.sh
```

— Hook paths, log paths, and CLI commands all named. product docs

06 **Projects: CLI creation and configuration** NEW 86

Projects gain full CLI lifecycle management: `amp projects create <repo>` creates a project from an existing GitHub or Git repository, `amp projects create --amp-hosted --name <name>` creates a blank Amp-hosted repo, and `amp projects create --push` (with `--remote <name>`) moves a local-only repo into an Amp-hosted project; `--personal` / `--workspace` control ownership. `amp projects update <project> --additional-repository <url>` (repeatable, up to 20) clones extra repos into every orb under `../repos/`, with `--clear-additional-repositories` to remove them, `--repository-url <url>` and `--repository-mode github|mapped` to change or migrate the primary repository, and running `amp projects update <project>` with no options refreshes GitHub metadata by stable ID. `amp projects delete` removes a project and archives its threads read-only, `amp clone owner/project-name` clones an Amp-hosted repo, and `--project` on `amp -ox` disambiguates among matching projects. Per-project settings add a Secrets & Env Vars tab (project values override workspace, personal overrides both), a Post-receive Webhook posting JSON after every push, Git Remote Aliases for matching mirror/SSH URLs, Orb Size as the default machine size, Git Identity (Amp Account or User Choice), and Public Read-Only Code Access for unauthenticated cloning; Puck can manage a project's name, repository URL/aliases, commit author, and orb size subject to caller permissions.

Bootstrap a new Amp-hosted project from an existing local repo and immediately push it so all orbs start from your current branch.

```
$ cd myproject
amp projects create --push
```

Wire a monorepo project to two additional repos so every orb checks out all three side-by-side under .. `/repos` /.

```
$ amp projects update acme/widgets \
  --additional-repository https://github.com/acme/api \
  --additional-repository https://github.com/acme/docs
```

Migrate an Amp-hosted project to an external GitHub repo after pushing the Git data there yourself.

```
$ amp projects update acme/widgets \
  --repository-url https://github.com/acme/widgets \
  --repository-mode github
```

— Every CLI flag and project setting enumerated. product docs

---

07 **Global hosted Plugins & Skills repositories** NEW 86

Hosted personal and workspace plugin/skill repositories are stored in Git and load everywhere Amp runs, not tied to one project or machine; workspace admins can publish shared plugins and skills via Amp prompts or the repository workflow. CLI support includes `amp plugins update <name>` and `amp skill update <name>` to pull updates, `amp plugins repositories` and `amp skills repositories` to list available repos and write access, and `amp clone user-plugins`, `amp clone user-skills`, `amp clone workspace-plugins`, `amp clone workspace-skills` for direct Git access. Project-scoped extensions live in `.amp/plugins/` and `.agents/skills/`, machine-local ones in `~/.config/amp/plugins/` and `~/.config/agents/skills/`; `amp plugins add <url>` installs machine-local by default, with `--target workspace` targeting the project's `.amp/plugins/` directory.

Check whether any imported plugins are outdated and update a specific one by name to keep your hosted extension current.

```
$ amp plugins update my-formatter-plugin
```

List all plugin repositories you have access to before cloning one for direct Git editing.

```
$ amp plugins repositories
amp clone user-plugins
```

Install a plugin scoped to the current project so it travels with its source code rather than being machine-local.

```
$ amp plugins add <url> --target workspace
```

— Names every CLI command and directory path involved. product docs

08 **Streaming JSON output for programmatic use** NEW 86

`--stream-json` (requires `--execute`) prints each conversation message as newline-delimited JSON for programmatic integrations and real-time monitoring; `--stream-json-input` reads a stream of user messages from stdin as newline-delimited JSON, including base64-encoded `image/jpeg`, `image/png`, `image/gif`, or `image/webp` blocks with an optional `source_path` field (omitted from stdout for Claude Code compatibility), and a `steer` boolean field delivers a queued message at the agent's next interruption point. `--stream-json-thinking` includes `thinking` and `redacted_thinking` content blocks in assistant messages, extending the schema beyond Claude Code compatibility.

Pipe Amp output into a script or log pipeline — get every turn as a parseable JSON object instead of plain text.

```
$ amp --execute "summarize the auth module" --stream-json | jq '.type, .message.content[0].text'
```

Include the model's reasoning trace in streaming output for debugging or audit logging of agent decisions.

```
$ amp --execute "review this PR for security issues" --stream-json --stream-json-thinking | jq 'select(.type=="assistant") | .message.content[] | select(.type=="thinking") | .thinking'
```

— Flags plus runnable jq pipelines demonstrate behavior. product docs

09 **Execute mode for non-interactive prompts** NEW 82

`-x` / `--execute` runs a single prompt non-interactively, prints the agent's final message, and exits; it auto-enables when stdout is redirected (e.g. `amp > response.txt`) and supports piped stdin alongside the prompt. `--fast` (alias for `--features fast`) enables Fast mode for a single invocation instead of the Standard-mode default, `--mcp-config` supplies an inline MCP server JSON without touching the settings file, `--executor runner:<id>` with `--runner-dir <absolute-path>` starts a thread on a named runner in a specific working directory, and the `AMP_API_KEY` environment variable (an `sgamp_` `-prefixed` token) authenticates non-interactive and CI/CD use instead of the short-lived session token from `amp login`.

Run a one-shot agent prompt in CI without an interactive session, authenticating via environment variable.

```
$ export AMP_API_KEY=sgamp_your-token-here
amp -x "Run the test suite and summarise any failures"
```

Attach a design file and requirements doc to an agent task running on a named remote runner.

```
$ amp -x "Implement this design" --executor runner:mac-mini --
attach ./design.png --attach ./requirements.txt
```

Use a one-off MCP server for a single query without touching your global settings file.

```
$ amp --mcp-config '{"everything": {"command": "npx", "args":
["-y", "@modelcontextprotocol/server-everything"]}}' -x "What
tools are available to you?"
```

— Flags and env var with runnable CI examples. product docs

---

10 **Orb Desktop environment** NEW 82

Orb threads gain a Desktop tab: a high-resolution interactive Linux (Debian 12) desktop where Amp can perform and record browser and GUI tasks. `--desktop-resume-script-file` on `amp projects create` / `amp projects update` sets a custom startup script from the CLI, and a committed `.agents/desktop/resume` file in the repo takes priority over the Project Settings value. `sudo -n /opt/amp-desktop/install-app <name>` installs built-in apps (`chrome` , `firefox` , `code` , `blender` , `terminal`), `apt-get` installs arbitrary graphical apps, and `sudo systemctl restart amp-desktop` reloads a modified resume script while a session is running.

Set a custom Desktop startup script for a new project so every orb opens Blender and LibreOffice instead of the default Chrome window.

```
$ amp projects create acme/widgets --desktop-resume-script-file
./desktop-resume.sh
```

Share the Desktop resume script with all contributors by committing it to the repository — it takes priority over the Project Settings value.

```shell
#!/usr/bin/env bash
set -euo pipefail
sudo -n /opt/amp-desktop/install-app blender
if ! pgrep -x blender > /dev/null; then
  blender &
fi
```

Install Visual Studio Code into the running orb Desktop and reload the session after updating the resume script.

```
$ sudo -n /opt/amp-desktop/install-app code && sudo systemctl
restart amp-desktop
```

— Named script paths, install command, and reload command. product docs

11 **Cross-client remote control and passkey enforcement** NEW

80

Cross-client access controls let workspace admins govern how CLI sessions are reached from elsewhere: the `--remote-control-terminal` / `--no-remote-control-terminal` CLI flags (overridable via the `AMP_REMOTE_CONTROL_TERMINAL` environment variable) enable or disable web Terminal access to a running CLI session, and the Enterprise Cross-Client Access control in Member Settings can prevent members from messaging CLI threads from the web or targeting runners via the `create_thread` tool, Puck, and automations. A Require Passkey Authentication for Web & App Interaction setting (with a workspace-level Enterprise enforcement option) requires recent passkey authentication before interacting with the agent on web or app.

Start a CLI session with web Terminal access enabled so a teammate or yourself can observe and interact with the agent's terminal from a browser or mobile device.

```
$ amp --remote-control-terminal
```

Enable web terminal access for all CLI sessions in a CI or shared environment without modifying scripts, by setting the environment variable instead of a flag.

```
$ export AMP_REMOTE_CONTROL_TERMINAL=1
amp
```

— Flag, env var, and settings names all given. product docs

12 **Configurable Ship Behavior** NEW 76

Ship Behavior is configurable per project (via `amp projects update` or the UI) with three options: default `Ship`, `Push to Branch` (commits and pushes a feature branch instead of the base branch, auto-reporting a pull request URL for GitHub repos), and `Custom Ship` (sends a user-written prompt up to 10,000 characters to the agent when Ship is pressed, e.g. `amp projects update acme/widgets --ship-behavior custom --custom-ship-prompt-file .agents/ship.md`, supporting merge queues, PR templates, or non-GitHub hosts). A `Restack` action in the thread Changes pane reorganizes an agent's intermediate commits into a cleaner set before shipping without pushing, and `Default Project Settings` under workspace Settings > Projects sets the default Ship Behavior for newly created projects.

Configure a project to use a custom Ship prompt stored in your repo, so delivery steps (branch naming, PR template, merge queue) are versioned like any other code.

```
$ amp projects update acme/widgets \
  --ship-behavior custom \
  --custom-ship-prompt-file .agents/ship.md
```

Pipe a generated or templated Ship prompt directly from stdout into the project update without writing a temporary file.

```
$ cat ./ship.md | amp projects update acme/widgets --ship-behavior custom --custom-ship-prompt-file -
```

— Named options and one CLI flag combination shown. product docs

13 **Prompting: thread references, feed filters, voice input** NEW 74

`@@` searches for and mentions a thread by ID or URL to pull its context into the current task, and threads can also be referenced directly by URL (e.g. `https://ampcode.com/threads/T-...`) or `@` `-prefixed` ID. `amp config vocabulary add <term>` registers words voice dictation mishears. Feed URLs support `time=` (`24h`, `72h`, `7d`, `all`) and a `q=` query syntax with `id:`, `label:`, `file:`, `parent:`, `project:`, `repo:`, `ref:`, `author:`, `visibility:` (`private` / `workspace` / `group` / `unlisted`), `archived:`, `snoozed:`, `pinned:`, `commit:` (boolean or SHA), `pr:` (boolean or number), `after:` / `updated_after:`, and `before:` / `updated_before:` (ISO dates or relative days/weeks), making feeds bookmarkable with precise scope. Voice input is available via `Cmd+Shift+D` (macOS) or `Ctrl+Shift+D` (elsewhere) to dictate-and-send; message queueing holds a mid-turn message until the agent finishes, triggered with `Shift+Enter` / `Ctrl+Enter` on web or `prompt: queue message` in the CLI, with default behavior set at Settings → Advanced

→ Default Message Send Behavior. The CLI also adds `thread: archive` to archive without removing from the web, `Tab` + `e` to edit a prior message, and support for mentioning skills by name in a prompt with no special syntax.

Register a domain-specific term that voice dictation consistently mishears, so future transcriptions get it right.

```
$ amp config vocabulary add Kubernetes
```

Bookmark a feed URL to review all workspace-shared threads labelled `bug` updated in the last 7 days.

```
$ /feed?q=label:bug%20visibility:workspace%20after:7d
```

— Every filter keyword named; mostly UI/feed syntax. product docs

14 Model routing, Dial modes, and specialist tools NEW

72

The Dial provides four built-in modes — `low`, `medium`, `high`, `ultra` — each combining a model, reasoning effort, system prompt, tools, and oracle for a difficulty level, and plugins can define custom modes (shown in the mode picker) and custom subagents (exposed as callable tools). Specialist subagents `Search`, `Oracle`, `Librarian`, and `Read Thread` each run in an isolated context window with file editing and terminal tool access; the `oracle` tool is a high-reasoning second-opinion model for complex debugging and code review, routed by agent mode (`high` / `medium` / `low` / `ultra`) and connected provider, and `Librarian` searches and reads code across public GitHub repositories and connected private repositories without leaving the session. The `Painter` tool (powered by GPT Image 2.5 Sunburst) generates and edits images, supporting up to 3 `@ -mentioned` reference images, transparent PNG output, and a `background: transparent|auto` option; `amp tools list` enumerates all built-in tools available in the environment. System models handle supporting jobs — analyzing images, PDFs, audio, and video; generating or editing images; naming threads; and compacting long conversation contexts — and Claude Fable models are supported via BYOK, configurable under `Model Routing` (per-key) or `Workspace Model Routing` (workspace-wide) after enabling data retention in the Anthropic Claude Console.

List every built-in tool Amp has access to in your current workspace.

```
$ amp tools list
```

Enable Claude Fable 5.1 for your team by wiring a data-retention Anthropic API key into Amp workspace model routing.

> 📍 In the Claude Console (`console.anthropic.com`), go to Settings › Workspaces, create or select a workspace, open it and go to Manage › Privacy controls, turn on data retention under Data retention , accept the confirmation, assign users, then create an API key. In Amp, add the key under Workspace Model Routing.

— Named tools and modes, mostly UI/config not CLI. product docs

15 **Keybindings and CLI editor customization** NEW 70

`amp.keymap` in `~/.config/amp/settings.json` (or `.amp/settings.json`) customizes CLI keybindings, supporting chords, arrays of bindings, and `null` to unbind defaults, and `amp config keymap` displays the full shortcut map. New bindings include Ctrl+V for clipboard image paste (with Ctrl+O → `prompt: paste image from clipboard` fallback), Ctrl+G to open the prompt in `$EDITOR` , Ctrl+R for prompt history, Alt+D to toggle reasoning effort, Alt+R to toggle fast mode, drag-and-drop image attachment from the file manager, Ctrl+J as a universal newline-insert that works in any terminal, and Shift+Enter for newline insertion in terminals that support modified Enter (Ghostty, WezTerm, Kitty, iTerm2, or tmux with `extended-keys`), enabled in tmux via `set -s extended-keys on` in `~/.tmux.conf` .

— Every keybinding named though no example commands given. product docs

16 **Puck personal assistant** NEW 68

Puck is a personal Amp assistant reachable via `Ctrl+/` on ampcode.com and the macOS app to start agents, find threads, manage projects, and coordinate parallel work. It's searchable in threads with the `puck:true` filter, opened in the TUI via `puck: open` (Ctrl+O then select, or Ctrl+X then `P`), triggered by a Talk to Puck Shortcuts app entry that starts dictation on iOS/macOS, reachable by sharing an image from any iOS app's share sheet, and reachable in Slack by mentioning `@Amp` in a channel or thread.

— Multiple named shortcuts and filters, mostly UI-driven. product docs

17 **Native macOS and iOS/iPadOS apps** NEW 66

Native macOS and iOS/iPadOS apps provide access to threads, orbs, and Puck with push notifications, plus per-device notification controls under App Settings (Mac: Amp menu > App Settings... ; iOS: sidebar > App Settings). The iOS composer gains Record Screen (with optional mic audio), Take Screenshot , and Attach Latest Screen Recording attachment options, and universal links make `ampcode.com/threads/...` open in the native app instead of Safari. macOS adds `File > New Amp Window` (Shift+Cmd+N) and shortcuts (Cmd+, , Option+Cmd+, , Cmd+[, Cmd+] , Cmd+R , Cmd++ , Cmd+- , Cmd+0), and up to five accounts can be signed in simultaneously per device.

— Concrete shortcuts and menu paths, no CLI equivalents. product docs

18 **CLI shell completions and IDE integration** NEW 66

`amp completions bash`, `amp completions zsh`, `amp completions fish`, and `amp completions pwsh` add tab completion for Bash 4+, Zsh, Fish, and PowerShell 7 with no separate Bun install or network connection; piped input (e.g. `echo 'commit all my changes' | amp`) is treated as the first interactive message, `amp update` manually triggers an update check and install, and `amp version` shows the installed version. The `ide connect` command palette entry (Ctrl+O) connects to VS Code, Cursor, Windsurf, Zed, and Neovim for open-file context and IDE-native editing with full undo support.

Enable Amp tab completion in Zsh so commands, flags, and subcommands autocomplete without a network connection or sign-in.

```
$ source <(amp completions zsh)
```

Run a task on a self-hosted runner instead of locally — useful for jobs requiring the runner machine's environment or credentials.

```
$ amp --executor runner:grandmas-garage-server
```

Feed a one-shot instruction into Amp from another tool or script without manually typing in the TUI.

```
$ echo "commit all my changes" | amp
```

— Completion commands and IDE targets named concretely. product docs

THINNER COVERAGE BELOW

19 **Space voice/video calls in threads** NEW 57

Space is a voice and video call built into every Amp thread via the Space tab, requiring no external meeting link. The Share Screen control lets any participant share a screen, window, or tab full-screen to others; the call keeps running in a floating window when navigating away from the thread pane. Active participants show in the sidebar's Multiplayer section, with a Space is live prompt offering Join and Dismiss to teammates with the thread open, and workspace admins can disable it entirely via the Enable Space toggle in Members settings.

— UI feature; navigation clear but no CLI/config surface. product docs

20 **Automations: scheduled agent threads** NEW 42

Automations let an Amp thread run on a one-time or repeating schedule driven by a plain-language prompt, with no cron expressions or separate workflow config. The

`get_schedule` tool lets the agent read the saved prompt at runtime, each scheduled thread shows an indicator with title, schedule, next-run time, and prompt plus pause/resume/delete controls, schedules can be changed by chatting with Amp in the same thread (e.g. 'Change this schedule to weekdays at 8:30 AM'), and a configured completion condition can clear a repeating schedule automatically once a monitoring task finishes.

— Described in prose; only one tool name given. product docs

One Runner, Many Worktrees (2026-09-22) BLOG ↗ LATEST STABLE

Amp runners can now spin up Git worktrees on demand and inject secrets/env vars into threads via `--amp-env`.

└──▷ TRY IT

Start a runner that automatically injects ampcode.com secrets into every new thread — useful when agents need API keys or tokens without hardcoding them.

```
$ amp --no-tui --runner-id mac-mini --discover-dirs --amp-env
```

Adds `--amp-env` flag to the `amp` CLI so runners fetch and inject workspace secrets and environment variables (personal, project, then workspace precedence) into each thread's shell commands, MCP servers, and plugins — without a runner restart.

Adds Git worktree creation to the runner directory picker: selecting New Worktree on any repository creates a sibling checkout (e.g. `~/code/amp-fix-flaky-login-test` on branch `fix-flaky-login-test`) and starts the thread there while leaving the main checkout untouched.

Adds an Archive and Remove Worktree thread action that archives the thread, runs `git worktree remove`, and deletes the branch when work in a runner-created worktree is complete.

Enables Puck to create worktrees programmatically by passing a worktree name to `create_thread`.

Adds New Directory and New Project options to the runner directory picker for paths that do not yet exist.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

CopilotKit OpenBot

Feedback ⓘ v0.0.15 NOTES ↗ RANK

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

> OpenBot's latest releases add an OAuth-backed Chat Completions proxy for Google and xAI model providers, email domain allowlisting for sign-in, and harden Kubernetes deployments by running computer pods as a non-root user, alongside fixes to MCP structured output, Google Drive shortcut handling, and LiteLLM provider/model parsing.

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

01 OAuth-backed Chat Completions proxy for model providers

95

BREAKING

Adds `OPENBOT_MODEL_OAUTH_FILE` environment variable to supply a Google or xAI OAuth credential file, enabling the server to mount `POST /api/model-provider/v1/chat/completions` as a local Chat Completions proxy that refreshes tokens automatically without exposing the provider's refresh token to callers. The endpoint handles streaming, tool calls and results, inline images, and function-call thought signatures via a translation layer bridging Google's native generation API to the Chat Completions contract. A provider 403 response no longer triggers a sign in again prompt — only a 401 does, so integrations relying on 403 for re-authentication will stop prompting users.

Proxy Chat Completions requests through the server using a Google or xAI OAuth grant instead of embedding an API key in the caller.

```
$ OPENBOT_MODEL_OAUTH_FILE=/run/secrets/openbot_oauth.json
openbot start
```

Send a streaming chat request through the local OAuth-backed proxy endpoint once the server is running.

```
$ curl -N -X POST http://localhost:<port>/api/model-
provider/v1/chat/completions \
  -H 'Authorization: Bearer <local-bearer-token>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gemini-2.0-flash", "stream": true, "messages":
[{"role": "user", "content": "Hello"}]}'
```

— Names endpoint, env var, mechanism, and includes a runnable curl example. v0.0.15

02 **Email domain allowlisting for sign-in** NEW 95

Adds `SIGNIN_ALLOWED_EMAIL_DOMAINS` environment variable (or `config.allowedEmailDomains` on the Helm chart) to restrict sign-in to specific email domains, using exact IDNA-normalised matching with no wildcards; a refusal writes a `session.refused` audit row. The server now refuses to start if `SIGNIN_ALLOWED_EMAIL_DOMAINS` is set while `MICROSOFT_OAUTH_TENANT_ID` is `common`, `organizations`, or `consumers` (a directory GUID must be set instead), or if the domain list resolves to an empty value (e.g. `@` or a stray `.`).

Lock an internet-facing OpenBot deployment to only your corporate email domain so that no outside accounts can sign in.

```env
SIGNIN_ALLOWED_EMAIL_DOMAINS=example.com
```

Apply the same domain restriction via Helm when deploying to Kubernetes.

```yaml
config:
  allowedEmailDomains:
    - example.com
```

— Names env var, Helm key, and gives runnable config examples. v0.0.14

03 **Non-root computer pods on Kubernetes** BREAKING 90

Computer pods on Kubernetes now run as `pwuser` (uid 1001) instead of root. On storage that cannot apply `fsGroup` (e.g. `hostPath` / rancher/local-path-provisioner on k3s), the computer pod refuses to start rather than silently misbehaving — operators must chown the directory or set `computers.podSecurityContext: null` before upgrading, and `helm upgrade --reuse-values` silently skips the new key so the value must be passed explicitly. In `computers.mode: sandbox`, existing Bots created before

the upgrade keep running as root; only newly created Bots run as `pwuser`, and the only way to migrate an existing Bot is `reset`, which deletes its volumes and logins.

— Details uid, config key, Helm flag, and the required migration path. v0.0.14

04 **BOT_PROVIDER and slash-containing model names in LiteLLM Bots** 80

BREAKING

The ADK, Strands, Agno, LlamaIndex, and CrewAI LiteLLM Bots now preserve the chosen `BOT_PROVIDER` in front of slash-containing model names (e.g. `qwen/qwen3-8b`) instead of treating the prefix as the provider. As a result, a `BOT_MODEL` value containing a slash (e.g. `bedrock/…`, `azure/…`, `openrouter/…`) no longer selects the provider from the prefix — the whole string is now the model name and `BOT_PROVIDER` governs the provider.

— Names five bots, both config keys, and the before/after behavior. v0.0.14

05 **EKS recipe IMDS hardening guidance** IMPROVED 60

The EKS cluster recipe now documents setting `disableIMDSv1` and `disablePodIMDS` on the node group to close pod access to the instance metadata service at 169.254.169.254.

— Names two node-group flags and the exact metadata IP. v0.0.14

THINNER COVERAGE BELOW

06 **Desktop setup OAuth sign-in and project bootstrap** NEW 45

Desktop setup now supports OAuth sign-in for Google Gemini and xAI alongside API-key authentication, and can create a CopilotKit project during setup. `desktop/PROVIDER_OAUTH.md` documents what a distributor must configure to enable Google or xAI OAuth sign-in in a custom build.

— Names the doc file but gives no command or config key. v0.0.15

07 **Fix for MCP structured output detection** IMPROVED 40

MCP tools that return answers in `structuredContent` with an empty content list are now read correctly instead of being reported as empty.

— Names the field but no reproduction steps or command. v0.0.14

08 **Google Drive shortcut resolution** IMPROVED 40

Google Drive shortcuts returned by search or recent files are now followed to their target file and read transparently.

— Describes fix behavior without a named API or flag. v0.0.14

09 **Desktop setup progress indicators** IMPROVED 30

Desktop setup now shows download progress in transferred bytes and elapsed time per step, with visual distinction between running, completed, and failed stages.

— UI description with no named flag or configurable surface. v0.0.15

10 **Pinned Bun version enforcement** IMPROVED 30

OpenBot now enforces its pinned Bun version on install and every start, acquiring its own copy when the installed or cached version does not match.

— Explains behavior but no version number or flag given. v0.0.15

└──▷ BREAKING ON UPGRADE

! A provider 403 response no longer raises a sign in again prompt — only a 401 does. Integrations that relied on a 403 triggering re-authentication will no longer prompt the user to sign in again.

! On the ADK, Strands, Agno, LlamaIndex, and CrewAI Bots, a `BOT_MODEL` value containing a slash (e.g. `bedrock/…`, `azure/…`, `openrouter/…`) no longer selects the provider from the prefix — the whole string is now the model name and `BOT_PROVIDER` governs the provider.

! On Kubernetes, computer pods now run as `pwuser` (uid 1001); on storage that cannot apply `fsGroup` (e.g. `hostPath` / rancher/local-path-provisioner on k3s), the computer pod will refuse to start rather than silently misbehave. Chown the directory or set `computers.podSecurityContext: null` before upgrading. Running `helm upgrade --reuse-values` silently skips the new key and keeps running as root — pass the value explicitly.

! In `computers.mode: sandbox`, existing Bot computers created before the upgrade continue to run as root (uid 0); only newly created Bots run as `pwuser`. The only way to migrate an existing Bot is `reset`, which deletes its volumes and logins.

! Setting `SIGNIN_ALLOWED_EMAIL_DOMAINS` (a non-empty domain list) while `MICROSOFT_OAUTH_TENANT_ID` is `common`, `organizations`, or `consumers` is now refused at start-up — a directory GUID must be set instead.

! A `SIGNIN_ALLOWED_EMAIL_DOMAINS` value that resolves to an empty list (e.g. `@` or a stray `.`) is now refused at start-up.

v0.0.15 NOTES ↗ LATEST STABLE

OpenBot v0.0.15 adds OAuth-based model provider auth via a new chat completions proxy endpoint and smarter desktop setup with dynamic port binding.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.0.15
https://github.com/CopilotKit/OpenBot.git
# already have the repo? check out this version:
$ git checkout v0.0.15
```

└──▷ TRY IT

Proxy Chat Completions requests through the server using a Google or xAI OAuth grant instead of embedding an API key in the caller.

```
$ OPENBOT_MODEL_OAUTH_FILE=/run/secrets/openbot_oauth.json openbot
start
```

Send a streaming chat request through the local OAuth-backed proxy endpoint once the server is running.

```
$ curl -N -X POST http://localhost:<port>/api/model-
provider/v1/chat/completions \
  -H 'Authorization: Bearer <local-bearer-token>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gemini-2.0-flash", "stream": true, "messages":
[{"role": "user", "content": "Hello"}]}'
```

Adds `OPENBOT_MODEL_OAUTH_FILE` environment variable to supply a Google or xAI OAuth credential file, enabling the server to mount `POST /api/model-provider/v1/chat/completions` as a local Chat Completions proxy that refreshes tokens automatically without exposing the provider's refresh token to callers.

Adds `POST /api/model-provider/v1/chat/completions` endpoint that handles streaming, tool calls and results, inline images, and function-call thought signatures when using a Google OAuth grant, with a translation layer bridging Google's native generation API to the Chat Completions contract.

Desktop setup now supports OAuth sign-in for Google Gemini and xAI alongside API-key authentication, and can create a CopilotKit project during setup.

Adds `desktop/PROVIDER_OAUTH.md` documenting what a distributor must configure to enable Google or xAI OAuth sign-in in a custom build.

Desktop setup now shows download progress in transferred bytes and elapsed time per step, with visual distinction between running, completed, and failed stages.

+1 more

└──▷ BREAKING ON UPGRADE

! A provider 403 response no longer raises a `sign in again` prompt — only a 401 does. Integrations that relied on a 403 triggering re-authentication will no longer prompt the user to sign in again.

v0.0.14 NOTES ↗ STABLE

OpenBot v0.0.14 adds email domain allowlisting, hardens single-user mode, and fixes MCP structured output and LiteLLM provider routing

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.0.14
https://github.com/CopilotKit/OpenBot.git
# already have the repo? check out this version:
$ git checkout v0.0.14
```

└──▷ USE IT

Lock an internet-facing OpenBot deployment to only your corporate email domain so that no outside accounts can sign in.

.env

```
SIGNIN_ALLOWED_EMAIL_DOMAINS=example.com
```

Apply the same domain restriction via Helm when deploying to Kubernetes.

yaml

```yaml
config:
  allowedEmailDomains:
    - example.com
```

Adds `SIGNIN_ALLOWED_EMAIL_DOMAINS` environment variable (or `config.allowedEmailDomains` on the Helm chart) to restrict sign-in to specific email domains, with exact IDNA-normalised matching and no wildcards; a refusal writes a `session.refused` audit row.

MCP tools that return answers in `structuredContent` with an empty content list are now read correctly instead of being reported as empty.

Google Drive shortcuts returned by search or recent files are now followed to their target file and read transparently.

The ADK, Strands, Agno, LlamaIndex, and CrewAI LiteLLM Bots now preserve the chosen `BOT_PROVIDER` in front of slash-containing model names (e.g. `qwen/qwen3-8b`) instead of treating the prefix as the provider.

The EKS cluster recipe now documents setting `disableIMDSv1` and `disablePodIMDS` on the node group to close pod access to the instance metadata service at 169.254.169.254.

└──▷ BREAKING ON UPGRADE

- ! On the ADK, Strands, Agno, LlamaIndex, and CrewAI Bots, a `BOT_MODEL` value containing a slash (e.g. `bedrock/…`, `azure/…`, `openrouter/…`) no longer selects the provider from the prefix — the whole string is now the model name and `BOT_PROVIDER` governs the provider.
- ! On Kubernetes, computer pods now run as `pwuser` (uid 1001); on storage that cannot apply `fsGroup` (e.g. `hostPath` / rancher/local-path-provisioner on k3s), the computer pod will refuse to start rather than silently misbehave. Chown the directory or set `computers.podSecurityContext: null` before upgrading. Running `helm upgrade --reuse-values` silently skips the new key and keeps running as root — pass the value explicitly.
- ! In `computers.mode: sandbox`, existing Bot computers created before the upgrade continue to run as root (uid 0); only newly created Bots run as `pwuser`. The only way to migrate an existing Bot is `reset`, which deletes its volumes and logins.
- ! Setting `SIGNIN_ALLOWED_EMAIL_DOMAINS` (a non-empty domain list) while `MICROSOFT_OAUTH_TENANT_ID` is `common`, `organizations`, or `consumers` is now refused at start-up — a directory GUID must be set instead.
- ! A `SIGNIN_ALLOWED_EMAIL_DOMAINS` value that resolves to an empty list (e.g. `@` or a stray `.`) is now refused at start-up.

WAS THIS USEFUL? 👍 👎

PydanticAI

Feedback ⓘ v2.47.0 NOTES ↗ RANK

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

> PydanticAI's biggest change is a wave of new ways to run and serve an agent — CLI, browser chat, AG-UI/Vercel AI streaming, ACP editor serving, A2A, realtime voice, and GitHub Agentic Workflows — plus uniform deferred-tool approval across all of them, alongside v2.47.0's typing improvements and two breaking changes to output routing and prompt content validation.

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

01 CLI and web chat interfaces for agents NEW

83

Any importable PydanticAI agent can now be run as an interactive terminal chatbot via agent.to_cli_sync() or the `clai -a <module>:<agent>` command, or launched with a built-in browser chat UI via agent.to_web() or `clai web`.

Expose any importable agent as an interactive terminal chatbot without writing a UI.

```
$ clai -a pydantic_ai_harness.coder:coder_agent
```

Launch a browser-based chat UI for a local agent during development or for team use.

```
$ clai web
```

– Exact runnable commands and function names given for both. product docs

02 Stricter validation of UserPromptPart content BREAKING

70

`UserPromptPart.content` now raises a clear error when it is not a `str` or a sequence, instead of silently iterating a `dict`'s keys; passing a `dict` as content previously sent the dict's keys silently and now raises an error instead.

– Exact field name and before/after behavior specified. v2.47.0

03 AG-UI and Vercel AI protocol streaming NEW

60

Agents can now stream runs to a custom frontend using the AG-UI protocol and the Vercel AI protocol, including support for Vercel's `useChat` React hooks.

– Names the protocols and hook, no setup snippet given. product docs

04 **Agent2Agent protocol support via fasta2a** NEW 60

Adds Agent2Agent (A2A) protocol support through the separate `fasta2a` package, letting an agent be served to other agents.

— Names the exact package but no usage snippet. product docs

05 **Output route for None renamed** BREAKING 60

A `None` output route is now named `None` instead of `NoneType`; any code that matches on the route name string `'NoneType'` will break.

— Exact old and new names given, no migration helper. v2.47.0

THINNER COVERAGE BELOW

06 **Realtime voice interface for agents** NEW 55

Agents gain a realtime voice interface allowing a live audio session using the same tools and observability as text runs, with conversation history handed off between voice and text runs.

— Explains handoff mechanism but names no API surface. product docs

07 **GitHub Agentic Workflows surface** NEW 55

Agents can now run on GitHub issues, pull requests, or a schedule, sandboxed on a runner with safe write-back outputs.

— Describes triggers and sandboxing, no exact workflow syntax. product docs

08 **Agent Client Protocol support for editors** NEW 50

Adds Agent Client Protocol (ACP) support so an agent can be served directly to Zed and other ACP-compatible editors.

— Names protocol and target editor, no command shown. product docs

09 **Deferred tool approval across all interfaces** IMPROVED 50

Deferred tools and approval now work uniformly across all agent interface surfaces — approval prompts in the CLI and approval UI events in frontends — from a single deployed agent.

— Names where approvals surface, no config key given. product docs

10 **Expanded field-type support in TypeSafeModel** IMPROVED

35

`TypeSafeModel` can now introspect and answer questions about three additional field types.

— The three field types are not named, limiting detail. v2.47.0

└──▷ BREAKING ON UPGRADE

! A `None` output route is now named `None` instead of `NoneType` — any code that matches on the route name string `'NoneType'` will break.

! Passing a `dict` as `UserPromptPart.content` now raises an error instead of silently sending the dict's keys as content.

v2.47.0 NOTES ↗ LATEST STABLE

PydanticAI v2.47.0 expands TypeSafeModel field-type support and tightens UserPromptPart content validation.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.47.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.47.0
```

Adds support for three additional field types that `TypeSafeModel` can introspect and answer questions about.

Raises a clear error when `UserPromptPart.content` is not a `str` or a sequence, instead of silently iterating a `dict`'s keys.

└──▷ BREAKING ON UPGRADE

! A `None` output route is now named `None` instead of `NoneType` — any code that matches on the route name string `'NoneType'` will break.

! Passing a `dict` as `UserPromptPart.content` now raises an error instead of silently sending the dict's keys as content.

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ harness-cli/v0.1.0 NOTES ↗

RANK

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

> Strands Agents' new harness CLI debuts with opus 5 and high thinking set as the default model and reasoning configuration.

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

01 Default model set to opus 5 with high thinking NEW

40

The harness CLI v0.1.0 ships with `opus 5` and `high thinking` enabled as the default model and reasoning configuration for the harness agent.

— Names default config but no flags or override mechanism given harness-cli/v0.1.0

harness-cli/v0.1.0 NOTES ↗ LATEST STABLE

Strands harness CLI v0.1.0 ships with opus 5 and high thinking enabled by default.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch harness-cli/v0.1.0 https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout harness-cli/v0.1.0
```

Sets opus 5 and high thinking as the default model and reasoning configuration for the harness agent.

WAS THIS USEFUL? 👍 👎

Nous Research Hermes

Feedback ⓘ v2026.9.21 NOTES ↗

CODE ↗ RANK

A terminal-native, self-improving AI agent for autonomous coding and tasks, with persistent memory, agent-created skills, and a multi-platform messaging gateway.

> Hermes shipped its public launch this window — a terminal-native agent with a curated, SHA-pinned plugin catalog, a 21+ platform messaging gateway, Bot Mode with group rooms, a Skills System, and multi-backend sandboxing — alongside a wave of CLI diagnostics (`hermes usage` , `hermes pause` / `resume` , `hermes mcp test`), gateway reliability monitoring, and security hardening around credential adoption, path traversal, and command allowlists.

└──▷ **WHAT SHIPPED · 63 FEATURES** most completely described first ? what's the number?

01 Plugin catalog install, update, and management commands NEW

98

Adds `hermes plugins install <name>` to install a reviewed catalog plugin at its pinned SHA, `hermes plugins enable <name>` to activate it, and `hermes plugins update <name>` to force-reinstall at a newer reviewed SHA while preserving enabled state and untracked files like `config.yaml` / `.env` (backed-up edits go to `~/.hermes/plugins-backup/<name>-<sha>/`). `hermes plugins list` (with `--json` and an `update_available` field) shows catalog installs as `catalog:<tier>@<sha>` ; `hermes plugins validate` runs a security scan used in catalog admission CI (`dangerous` blocks entry, `caution` surfaces to reviewers); `hermes plugins remove <name>` and `--allow-removed` handle removal and reinstatement from the `plugin-catalog/removed.yaml` blocklist. `--ref <sha>` records the exact checked-out SHA in `~/.hermes/plugins/.install-metadata.json` . The catalog YAML entry format (`plugin-catalog/<name>.yaml`) defines `name` , `repo` , `sha` , `tier` , `category` , `maintainer` , `capabilities` , `requires_hermes` , `platforms` , `docs_url` , `version` , `image` , `screenshots` , and `readme` . v2026.9.21 adds a `plugins.manage` remove action in the gateway and a one-click Uninstall button on the Desktop Plugins hub.

Add Snyk scanning capability to Hermes — code, dependency, container, and IaC — in two commands using the reviewed catalog entry.

```
$ hermes plugins install snyk
hermes plugins enable snyk
```

Check which catalog plugins have updates available, then update a specific one to its newly reviewed SHA.

```
$ hermes plugins list --json
hermes plugins update snyk
```

Reinstate a plugin that was added to the removed list when you have explicitly accepted the risk.

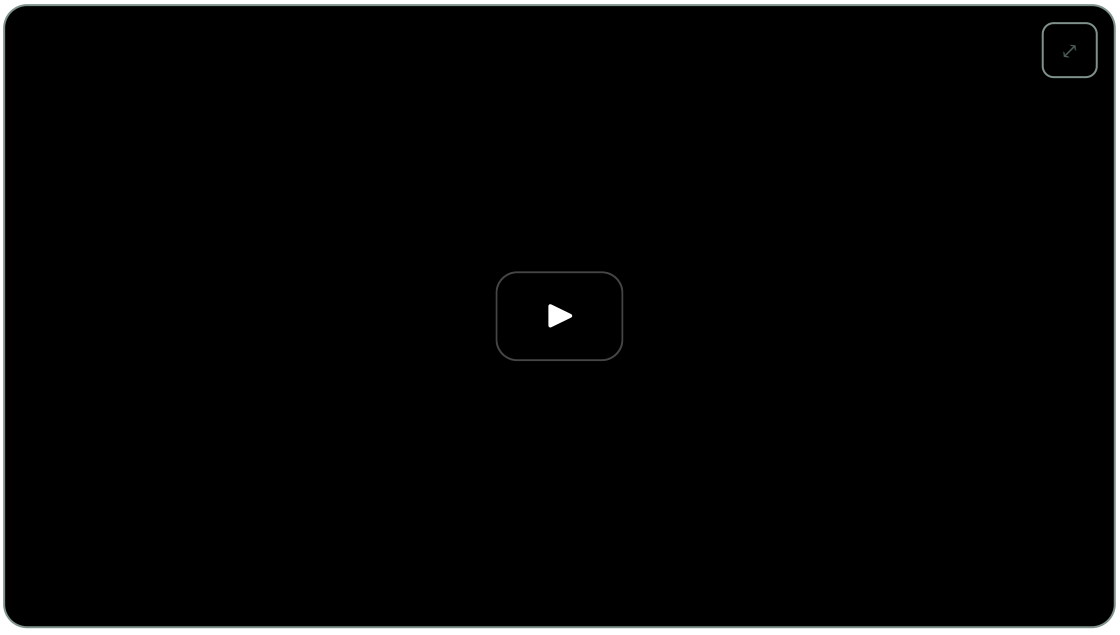
```
$ hermes plugins install --allow-removed <name>
```

— Names every command, flag, and config path with mechanism. v2026.9.21

02 **`config get`/`config set` with masking, routing, and validation**

NEW 93

Adds `config get <key> [--json] [--raw]` with automatic masking of credential-shaped values (`api_key` , `*_TOKEN` , `*_SECRET` , `password`) in output — `--raw` or `security.redact_secrets: false` prints the real value. Adds `config set <key> <value> [--force]` which routes `UPPER_SNAKE_CASE` values to the environment-variable file (dotted keys go to the main config file), type-checks list/mapping keys (`custom_providers` , `model.aliases` , `display.platforms` , `plugins.enabled` , `model_catalog.excluded_providers`) against the schema, gives did-you-mean hints on wrong-prefix keys, and refuses denylisted names like `HERMES_YOLO_MODE` and `PATH` ; `--force` overrides the did-you-mean warning.



— Names both commands, every flag, and the exact denylist/schema keys. product docs

03 **Gateway event-loop watchdog and heartbeat monitoring** NEW

93

Adds `gateway.loop_watchdog_probe_interval_s` (default 30s) and `gateway.loop_watchdog_max_strikes` (default 3) to tune the built-in asyncio event-loop liveness watchdog, which dumps thread stacks, stamps `gateway_state.json` with `gateway_state: degraded` / `exit_reason: loop_liveness_watchdog`, and exits for supervisor restart when the loop freezes; `gateway.loop_watchdog: false` disables it entirely. Heartbeat monitoring via `gateway_state.json`'s `updated_at` field (refreshed every 60s) shows a Heartbeat stale dashboard badge and emits a warning when the stamp is over 120s old. On-demand stack dump via `SIGUSR2` (`kill -USR2 <gateway pid>` on Linux/macOS) appends all thread stacks to `~/.hermes/logs/gateway_faulthandler.log` without restarting the gateway.

Inspect what a stalled gateway is doing without taking it offline — dumps all thread stacks to the log on demand.

```
$ kill -USR2 $(pgrep -f 'hermes.*gateway run')
```

Tune the watchdog to tolerate a slower loop (60 s probes, 5 misses) and suppress noisy warning notifications on all surfaces except Telegram.

```yaml
gateway:
  loop_watchdog_probe_interval_s: 60
  loop_watchdog_max_strikes: 5

suppress_warning_notifications: true

telegram:
  suppress_warning_notifications: false
```

— Names every config key, file, signal, and threshold. product docs

04 **Group room management and stop-directive controls** IMPROVED

90

Adds Manage members (room header people icon or Group settings) to add/remove multiple members while preserving room history and per-member sessions; a Compress history button per member runs context compression against that member's hidden per-room session, equivalent to `/compress` but targeting sessions not reachable from the sidebar; a Detect stop directives toggle lets rooms disable positional stop parsing while keeping the Stop button active; `@all` / `@everyone` releases all held members, and positional stop words (`stop @bot` , `@bot please pause`) hold individual members without treating prose occurrences as directives. Adds room picture upload/generation via New Group Chat and Group settings ; Open recent session (right-click a Bot) opens its most recent conversation as a tab beside the Bot Chat; Pin to top keeps a room above unpinned items per Desktop; rooms can be filed into sections via right-click or

drag. Room state now lives in the install's root `shared-state.db` (pre-existing Group Chats copy over automatically); unresolved member failures remain visible in the Activity summary with secret-shaped tokens redacted; renaming and disbanding a room are now durable across all clients.

When a group room member starts returning empty replies because its hidden session has grown too large, compress that member's context without touching the room log.

> 📍 In the room header, open Group settings → find the member row → click Compress history. The dialog reports how many messages were folded. Repeat for each affected member separately.

— Names every UI control and the underlying storage change. product docs

---

05 **`hermes usage` for rate-limit windows** NEW 90

Adds `hermes usage [--provider NAME] [--json]` to fetch the configured account's rate-limit windows (Codex 5-hour/weekly, Anthropic OAuth, OpenRouter credits) without starting a session, returning a stable JSON schema with `provider`, `source`, `title`, `plan`, `fetched_at`, `windows` (`label`, `used_percent`, `resets_at`, `detail`), `details`, and `unavailable_reason`.

Check Codex and Anthropic rate-limit windows from a cron job or shell script without starting an interactive session.

```
$ hermes usage --provider openai-codex --json | jq '.windows[] | {label, used_percent, resets_at}'
```

— Exact command, flags, and full JSON schema named. product docs

06 **Per-bot voice, model, and OAuth sign-in** IMPROVED 88

Adds `tts.*` per-Bot voice settings in each Bot's profile config so different Bots sound distinct even on the same gateway; supports a per-Bot model (any provider/model pair) overriding the launch profile so different Bots can run on different models side by side; adds `hermes -p <name> auth add <provider>` to sign a Bot into single-use OAuth providers (Anthropic, OpenAI Codex, xAI) that cannot be copied from the main profile; and adds `sessions.auto_archive` governing the automatic idle-archive sweep, which never retires a Bot Chat — only an explicit archive does.

Sign a Bot into an OAuth provider whose credentials cannot be copied from the main profile, so it can make authenticated API calls under its own identity.

```
$ hermes -p summarizer-bot auth add anthropic
```

— Names config keys and a runnable command with scope. product docs

07 **Gateway systemd and launchd service integration** NEW 88

Maps `systemctl reload hermes-gateway` to `SIGUSR1` in the installed systemd unit, triggering a graceful drain, process exit, and supervisor relaunch; adds `ExecStop=` so `systemctl stop` / `restart` record a planned-stop marker and persist `gateway_state=stopped` without a `Failed with result exit-code` journal entry. Adds `hermes gateway install --system --run-as-user <user>` to install the system-scope systemd unit inside containers where a user-scope install would conflict with a host `systemd --user`. Routes the launchd-managed gateway through `/usr/bin/osascript` (`do shell script 'exec …'`) on macOS so child processes inherit osascript's application identity and aren't blocked by Local Network Privacy, fixing `No route to host` errors for LAN devices.

Install the Hermes gateway as a system-scoped service inside a container (e.g. podman/distrobox) to avoid a second gateway process starting on the host.

```
$ sudo hermes gateway install --system --run-as-user myuser
```

— Names exact commands, signals, and the macOS fix mechanism. product docs

08 **Multi-platform installer with headless deployment flags** NEW 86

Installs on Linux, macOS (Apple Silicon), Windows (native and WSL2), and Android (Termux) via `curl -fsSL https://hermes-agent.nousresearch.com/install.sh | bash` or a PowerShell equivalent, auto-installing Python 3.11 (via uv), Node.js v26, ripgrep, and ffmpeg with no manual prerequisites. `--skip-browser` bypasses

Playwright/Chromium installation for headless deployments, and `--skip-computer-use` defers `cua-driver` installation to on-demand activation. `HERMES_HOME` overrides the data directory location, including symlinked homes and external/NAS volumes, and a root-mode FHS layout (`/usr/local/lib/hermes-agent/` , `/usr/local/bin/hermes`) supports shared-machine and system-service deployments.

Stand up Hermes on a headless server (no browser automation needed) for a system service account, skipping Playwright and Computer Use installation.

```
$ curl -fsSL https://hermes-agent.nousresearch.com/install.sh | bash -s -- --skip-browser --skip-computer-use
```

Bootstrap Hermes on a fresh Linux/macOS/WSL2 machine and start an interactive session immediately.

```
$ curl -fsSL https://raw.githubusercontent.com/NousResearch/hermes-agent/main/scripts/install.sh | bash
```

— Exact install command and every flag/env var named. product docs

---

09 **Controls for adopting external CLI credentials** NEW

86

Adds `auth.adopt_external_logins` config key (default `true`) to control whether Hermes automatically borrows and refreshes Codex CLI's `~/.codex/auth.json` and Claude Code's `~/.claude/.credentials.json` ; setting it to `false` removes those entries from `hermes auth list` and stops automatic token adoption. Adds `hermes auth add anthropic` and `hermes auth add openai-codex` to manually import provider logins without relying on automatic adoption.

Prevent Hermes from borrowing and silently refreshing your Codex CLI or Claude Code tokens when running those CLIs alongside Hermes.

```yaml
auth:
  adopt_external_logins: false
```

Import your own Anthropic login so Hermes stops competing with Claude Code over a shared rotating refresh token.

```
$ hermes auth add anthropic
```

— Named config key, default, and two runnable commands. product docs

10 **`hermes codex-runtime migrate` for Codex config** NEW

86

Adds `hermes codex-runtime migrate [--dry-run] [--json]` to regenerate the Hermes-managed block in `~/.codex/config.toml` for a selected profile without a chat session; `--dry-run` computes and reports the migration without writing the file, `--json` prints the full report (`migrated` , `preserved_user_servers` , `skipped_keys_per_server` , `target_path` , `written`) as JSON.

Preview what the Codex config migration will change before committing it to disk.

```
$ hermes -p default codex-runtime migrate --dry-run --json | jq
'{migrated, preserved_user_servers, target_path}'
```

— Exact command, flags, and JSON report fields with example. product docs

11 **Skills System with agent-managed skills and auto-load** NEW

85

Introduces a Skills System with on-demand knowledge documents, progressive disclosure, agent-managed skills, and a Skills Hub, with background maintenance via a Curator doing usage tracking, staleness detection, archival, and LLM-driven review. Adds `skills.auto_load` config key to pin skills into every new session's prompt. Adds `--ignore-rules` flag (and `HERMES_IGNORE_RULES=1` env var) to suppress missing/disabled skill warnings during auto-load; more broadly it skips auto-load along with `AGENTS.md` , `SOUL.md` , `.cursorrules` , and memory injection while still honoring explicit skill loads.

Pin the github pr workflow skill so it loads automatically at the start of every session without manual invocation.

```yaml
auto_load:
  - my workflow
  - github pr workflow
```

Ensure the same skills load automatically for every new Hermes session across all session types without needing to activate them manually each time.

```yaml
skills:
  auto_load:
    - dev
    - github
    - pr
    - workflow
```

Start a session with all rule and memory injection suppressed — useful for a clean, minimal context when auditing or testing model behavior.

```
$ hermes --ignore-rules
```

Pin a skill into every new session so it is always available without manual loading.

```yaml
skills:
  auto_load:
    - threat-intel-lookup
```

— Names config key and flag with concrete behavior and scope. v2026.9.21

12 **Windows native execution and Task Scheduler gateway** NEW

83

Supports native Windows 10/11 execution including UTF-8 console, Git Bash, and gateway-as-Scheduled-Task. Adds Windows Task Scheduler support via `hermes gateway install`, `hermes gateway start`, `hermes gateway stop`, and `hermes gateway status`, with a generated `wscript.exe` / `.vbs` launcher under `%USERPROFILE%\\.hermes\\gateway-service\\`, `RestartOnFailure` and logon `Delay` policies; `hermes gateway status` detects and reports task registration drift.

— Names commands and file paths with concrete policies. product docs

13 **`hermes mcp test` for MCP server health checks** NEW

83

Adds `hermes mcp test <server>` subcommand that connects to an MCP server, lists discovered tools, and exits with distinct codes — success on completed connect, failure when the connection fails, and argparse usage error when the server is not in config — so health probes and cron watchdogs can branch on `$?` without parsing output.

Use in a cron watchdog to alert when an MCP server becomes unreachable, branching on exit code instead of scraping output.

```
$ hermes mcp test my-server || echo 'MCP server unreachable' |
mail -s 'Hermes MCP alert' ops@example.com
```

— Exact command with exit-code contract and runnable example. product docs

14 **Memory configuration, staging, and per-profile scoping** NEW

83

Adds `max_input_tokens` config key under `auxiliary:` to cap total replayed input tokens consumed by a single background review loop (`<= 0` unlimited; default derived from 75% of the review model's context window, capped at 600,000 tokens or 120,000 when the window can't be resolved). Clarifies that memory writes outside the interactive CLI are staged and require `/memory approve all` before reaching the file; that memory is scoped per profile (e.g. `hermes -p work` reads from `~/.hermes/profiles/work/memories/` rather than the default `~/.hermes/memories/MEMORY.md`); and that `memory.memory_enabled: false` removes the memory tool entirely.

Limit background review token spend on a small local model to avoid runaway costs across tool iterations.

```yaml
auxiliary:
  max_input_tokens: 49152  # e.g. 75% of a 65,536-token local
model
```

Approve all staged memory writes queued outside the interactive CLI so they actually persist to disk.

```
$ /memory approve all
```

— Names config keys, defaults, and file paths with examples. product docs

15 **Session database journal mode migration** NEW

83

Adds `hermes sessions set-journal-mode delete` to convert an existing WAL-mode SQLite session database to delete journal mode safely, with process-hold detection and header verification; emits `<db> is in WAL mode despite database.journal_mode=delete` and names the blocking process by PID and command when the mode can't be downgraded live.

Convert an existing WAL-mode database to delete journal mode when moving a profile to a network mount where WAL is unsafe.

```
$ hermes sessions set-journal-mode delete
```

— Exact command with safety checks and a runnable example. v2026.9.21

16 **`--format stream-json` with distinct exit codes** NEW

83

Adds `--format stream-json` for structured JSONL one-shot output, with exit codes distinct from `chat -q` for completed turn, failed/partial/iteration-budget exceeded, interrupted, and a completed turn with no text at all; a Kanban worker with `HERMES_KANBAN_TASK` set exits with a dedicated code on provider rate-limit/5xx/quota failure so the dispatcher requeues without counting a failure.

Consume structured agent progress in a pipeline without scraping terminal output — pipe JSONL events to jq to extract only tool results.

```
$ chatgpt-codex --query 'Inspect this repository' --format
stream-json 2>/dev/null | jq 'select(.type=="tool_result") |
{output, duration_ms, is_error}'
```

Stream a query's response as newline-delimited JSON for pipeline ingestion or real-time log forwarding.

```
$ hermes --format stream-json 'summarize the last 24 h of
alerts'
```

— Exact flag, exit-code contract, and runnable pipeline example. v2026.9.21

17 **`hermes pause`/`hermes resume` global stop** NEW

80

Adds `hermes pause` and `hermes resume` as a global emergency stop that halts all new cron fires (built-in ticker, managed-cron webhook, misfire catch-up), kanban dispatch, and gateway turns without killing in-flight work; a paused recurring slot that comes due stays due and fires a single catch-up run or a logged skip on resume.

— Exact commands with defined resume semantics. product docs

18 **Post-install configuration and lifecycle commands** NEW

78

Adds `hermes model`, `hermes tools`, `hermes gateway setup`, `hermes config set`, `hermes config get`, and `hermes setup` for post-install configuration; `hermes doctor` reports missing dependencies, storage problems, and detected install method; `hermes desktop` launches the desktop app after a CLI-only install; `hermes`

`import` / `hermes profile import` restore a full backup or migrate a single agent profile; `hermes backup` / `hermes profile export` export configuration and profiles; `hermes config check` / `hermes config migrate` detect and repair missing config after an update; `hermes update` auto-detects the install method (git, Docker, or NixOS) and prints the matching update command.

Migrate a full Hermes configuration to a new machine, then separately bring over a single agent profile.

```
$ hermes import
hermes profile import
```

— Names each command but gives little mechanism per one. product docs

19 **`--usage-file` token accounting for one-shot runs** NEW 78

Adds `--usage-file` support (e.g. `--usage-file ~/.hermes/cache/scratch/usage.json`) to one-shot runs so full token accounting — including `auxiliary.by_task` per-task breakdown and `total_including_auxiliary.estimated_cost_usd` — is written to a file queryable with `jq`.

— Exact flag, file fields, and a queryable output path. product docs

20 **Extensible plugin system and gateway event hooks** NEW 73

Provides an extensible plugin system for custom tools, hooks, and integrations, plus built-in lifecycle-hook plugins (disk-cleanup and others). Adds trusted-by-placement Gateway event hooks: any directory placed at `<profile home>/hooks/<name>/` containing `HOOK.yaml` and `handler.py` is automatically loaded in-process at gateway startup via HookRegistry.discover_and_load() — no `plugins.enabled` entry or prompt required.

— Names the exact directory layout and loader mechanism. product docs

21 **IPv6 route check in `hermes doctor`** NEW 73

Adds an IPv6 route check to the API Connectivity section of `hermes doctor`, opening a 2-second IPv6 TCP connection to a known dual-stack host and warning (naming the remedy `network.force_ipv4: true`) when the prefix is blackholed; no IPv6 route at all is reported as OK, and the check is skipped when `force_ipv4` is already set.

— Names the exact remedy config key and check behavior. product docs

22 **Network exemption config keys for proxies and local hosts** NEW 72

Adds `fake_ip_ranges` config key to exempt TUN-proxy sentinel IP blocks (e.g. `198.18.0.0/15`) from the private-address guard, unblocking outbound fetches on hosts running Mihomo/Clash `fake-ip` or Surge enhanced mode. Adds `security.allow_private_urls: true` config key required for image-generation providers hosted on a local network to cache result URLs locally.

Allow outbound fetches to succeed on a host running Mihomo/Clash in fake-ip mode, where DNS returns 198.18.x.x sentinel addresses for every proxied name.

```yaml
security:
  fake_ip_ranges:
    - 198.18.0.0/15
```

— Two named config keys with concrete example and use case. product docs

23 **Hermes Desktop app and refinements** NEW 71

Ships a native desktop app for macOS, Windows, and Linux with streaming tool output, side-by-side previews, a file browser, voice, cron, profiles, skills, and settings. Adds `desktop.font_family` (editable at Settings → Appearance → Chat Font) to set the chat/interface font to any installed family (e.g. `OpenDyslexic`, `Atkinson Hyperlegible`) or CSS font stack. v2026.9.21 adds one-click local engine updates from the Desktop, and the Desktop now attaches to the running host backend instead of spawning a second one.

— Names the config key and UI path; rest is prose feature list. v2026.9.21

24 **Windows device-namespace path blocking** NEW 71

Blocks Windows NT/device-namespace paths (`\\??\\...`, `\\\\.\\...`, `\\\\?\\UNC\\...`, `\\\\?\\GLOBALROOT...`) on all platforms for both reads and writes across `read_file`, `search_files`, `@file:`, `@folder:` context references, and the ACP file bridge.

— Names exact path patterns and every affected surface. product docs

25 **Import from Claude Code and Codex CLI** NEW 70

Imports Claude Code (`~/.claude`) or OpenAI Codex CLI (`~/.codex`) setups — including instructions, allowlists, MCP servers, skills, and memories — with a single command, via the CLI's Import from Other Agents flow.

Migrate an existing Claude Code environment into Hermes, preserving instructions, allowlists, MCP servers, skills, and memories.

📍 In the Hermes CLI, run: Import from Other Agents and point it at `~/.claude` to pull in your Claude Code setup.

— Names source paths and scope; UI path actionable, not a flag. product docs

26 **Plugin catalog website and community plugin listings** NEW 68

Publishes the catalog as a live JSON document at `https://hermes-agent.nousresearch.com/docs/api/plugin-catalog.json` (fetched by search/install/update at most every six hours, cached under `~/.hermes/cache/` with a 24-hour offline fallback), plus a browsable visual catalog at `/docs/plugins` with category shelves (Memory, Desktop, Platforms, Web & Browser, Tools, Voice, Automation, Models), search, tier filters, capability chips, and copyable install commands. Adds per-plugin pages at `/docs/plugins/<name>` and per-author pages at `/docs/plugins/by/<maintainer>`, screenshot and README entry fields, pinned-commit READMEs with added/updated sorting, and a bundled `snyk` catalog entry pinning the Snyk CLI (`npx -y snyk@<version> mcp`) with the `snyk-security-scan` skill. v2026.9.21 adds a dozen community plugins: tailscale, ssh, shodan, terminal, rss, resetwatch, done-bell, kiwi, cognee, and Octen.

— Lists concrete pages and plugins but describes UI, not a command. v2026.9.21

27 **Configurable compression stream timeout** NEW 66

Adds `auxiliary.compression.no_progress_timeout` config key (default `60s`) to abort a stalled Responses-API summary stream after a configurable silent gap, independently of `auxiliary.compression.timeout`.

Widen the silent-gap tolerance for a long reasoning summary running over a Responses-API stream so it is not aborted prematurely.

```yaml
auxiliary:
  compression:
    no_progress_timeout: 180  # tolerate a 3-minute silent gap
on a long reasoning summary
```

— Named key with default and runnable config example. product docs

28 **Configurable MCP discovery concurrency** NEW 66

Adds `mcp.discovery_concurrency` config key (default 4, `0` = unlimited) to control how many MCP servers are contacted in parallel during discovery.

Throttle MCP discovery to avoid overwhelming many servers at once, or lift the cap entirely in a high-trust environment.

```yaml
mcp:
  discovery_concurrency: 8
```

— Named key, default, and a runnable config example. v2026.9.21

29 **One-step Nous Portal onboarding** NEW 65

`hermes setup --portal` logs in, sets Nous as the provider, and enables the Tool Gateway (web search, image generation, TTS, cloud browser) in a single command.

One-command onboarding via the Nous Portal to get 300+ models and the Tool Gateway without managing individual API keys.

```
$ hermes setup --portal
```

— Exact command given but mechanism is brief. product docs

30 **LSP-based semantic diagnostics** NEW 65

Supports LSP-based semantic diagnostics wiring real language servers — `pyright`, `gopls`, `rust-analyzer`, and others — into the post-write lint check used by `write_file` and `patch`.

— Names tools and integration point but no config shown. product docs

31 **Gateway controls for unauthorized senders** NEW 65

Adds `gateway.allow_all_users: true` (also accepted as top-level `allow_all_users: true`) config key, bridged to an env var at gateway startup and re-derived on every config reload. Adds `unauthorized_dm_decline_message` config key to customize the text sent when an unauthorized sender messages the gateway, alongside a new `decline` behavior for handling unauthorized DMs.

— Names two config keys but no worked example. v2026.9.21

32 **Secret-store path protection under HERMES_HOME** NEW 65

Protects Hermes secret-store paths — `auth/google_oauth.json`, `cache/bws_cache.json`, `cache/bws_cache.enc.json`, `vault/`, `browser-profile/` under `HERMES_HOME` — with read-deny while keeping them writable.

— Names every protected path with the exact access rule. product docs

33 **Configurable warning notification suppression** NEW 64

Adds `suppress_warning_notifications` config key — settable globally or per messaging surface — to opt out of automatic engine warnings, retry/fallback diagnostics, watchdog notices, cron/Kanban failure notifications, and adapter-generated error notices.

Tune the watchdog to tolerate a slower loop (60 s probes, 5 misses) and suppress noisy warning notifications on all surfaces except Telegram.

yaml

```yaml
gateway:
  loop_watchdog_probe_interval_s: 60
  loop_watchdog_max_strikes: 5

suppress_warning_notifications: true

telegram:
  suppress_warning_notifications: false
```

— Named key with per-surface scope and a config example. product docs

34 **Terminal UI with live work monitor** NEW 62

Provides an Ink-based TUI with mouse support, rich overlays, and non-blocking input, plus a full-screen live work monitor for subagents and background processes with arrow-key navigation, reachable without losing the composer draft. A toggle switches the dock between multi-row preview and a single summary line without moving composer focus, and composer path completion follows the configured `terminal.backend` (including `ssh` and `docker`) while `!` commands always run on the local host.

— Describes UI mechanics but no direct commands to run. product docs

35 **Session heartbeats and recurring loops** NEW 62

Adds Session Heartbeats — a recurring prompt (`/heartbeat`) that re-enters an idle session on a configurable interval — and Recurring Loops to re-run a prompt on a recurring interval inside a session.

Keep a long-running deployment session alive by injecting a check prompt every 10 minutes when idle.

```
$ /heartbeat every 10m Check the deployment.
```

— Names the command with a runnable example. product docs

36 **Tirith scanner false-positive downgrades** IMPROVED 60

Downgrades two Tirith false positives — `lookalike_tld` warnings targeting only the `.app` gTLD and `variation_selector` warnings for U+FE0F emoji suffixes — to `allow` so they never prompt or block in cron.

— Names the two specific checks changed, no toggle given. product docs

THINNER COVERAGE BELOW

37 **Context files and inline context references** NEW 59

Automatically injects context files `.hermes.md`, `AGENTS.md`, `CLAUDE.md`, global `SOUL.md`, and `.cursorrules` into every conversation, and provides inline `@ -` `syntax` (Context References) for attaching files, folders, git diffs, and URLs directly into messages.

— Names the files and syntax but no worked example. product docs

38 **Credential masking in allowlist output** NEW 57

Masks credentials (`ghp_…`, `bot<id>:<token>`, `KEY=value` assignments, bearer tokens) in allowlist `allow` examples and command payloads using the same redactor as terminal output; masked text is never used as an allowlist glob pattern.

— Names exact token formats masked, no toggle to control it. product docs

39 **Dangerous-pattern allowlist rules in unattended sessions** NEW 56

Adds `dangerous-pattern rule key` support to the command allowlist so entries like `script execution via heredoc` are honored on every surface including unattended `cron` and webhook sessions running under `unattended_mode: deny`.

— Names the rule key and mode but no full config example. product docs

40 **NixOS module and Nix flake support** NEW 53

Provides a Nix flake, declarative NixOS module, and optional container mode for Nix/NixOS users via the `nix-setup` guide, plus `nix run` support for declarative deployment.

— Names the artifacts but no example invocation. product docs

41 **Configurable terminal sync-back size cap** NEW 53

Adds `terminal.sync_back_max_bytes` config key (bytes) to raise the 2 GiB default cap on remote state archives during sync-back.

— Names the key and default but no worked example. product docs

42 **Document extraction via read_file** NEW 51

Provides Document Extraction via `read_file` to convert PDFs, Office documents, and notebooks to text.

— Names the tool and formats but no options. product docs

43 **Multi-platform messaging gateway** NEW 50

Adds a messaging gateway spanning 21+ platforms — 19 natively plus IRC and Microsoft Teams via plugins — including Telegram, Discord, Slack, SMS, and Matrix.

— Names platforms but no setup steps given here. product docs

44 **Kanban multi-gateway deployment** NEW 50

Provides Kanban multi-gateway deployment: one kanban board across several per-profile gateways with a single dispatcher and profile-owned delivery, plus worker lanes.

— Describes architecture but no config or command. product docs

45 **Bot Mode for named autonomous bots** NEW 48

Delivers Bot Mode: named bots with their own chat, role, model, memory, skills, and avatar, capable of running routines, sharing group chats, and messaging each other.

— Describes capability but no config keys or commands. product docs

46 **Web dashboard customization** NEW 48

Supports extending the web dashboard with custom themes, plugins, palettes, typography, layouts, custom tabs, shell slots, page-scoped slots, and backend API routes.

— Lists extension points but no API detail. product docs

47 **External memory provider integrations** NEW 45

Integrates external memory providers including Honcho, OpenViking, Mem0, Hindsight, Holographic, RetainDB, ByteRover, and Supermemory.

— Names eight providers but no setup steps. product docs

48 **Google Workspace skills** NEW 45

Supports Google Workspace skills (Gmail, Calendar, Drive, Sheets, Docs) via OAuth2-authenticated Google APIs.

— Names apps and auth method but no command. product docs

49 **Managed Scope for admin-pinned config and secrets** NEW 43

Supports Managed Scope — administrator-pinned, user-immutable config and secrets via a system-level managed directory.

— Describes purpose but no directory path or example. product docs

50 **Deliverable Mode for generated files** NEW 41

Delivers Deliverable Mode, shipping generated charts, PDFs, spreadsheets, and other files as native attachments in messaging platforms.

– Describes outcome but no mechanism or command. product docs

51 **Fail-closed on malformed shell quoting** NEW 41

Fails closed with a malformed executable payload error when a command's shell quoting cannot be parsed, preventing quoting tricks from hiding subsequent commands.

– Describes the guard but no command or config to invoke it. product docs

52 **SOUL.md agent personality and system prompt** NEW 40

Defines agent personality and system prompt via `SOUL.md`, with support for built-in personalities and custom persona definitions.

– Names the file but no mechanism or examples given. product docs

53 **Mixture of Agents model presets** NEW 40

Enables Mixture of Agents (MoA) presets that appear as selectable models under the Mixture of Agents provider.

– Brief description, no configuration example. product docs

54 **Multi-backend agent sandbox execution** NEW 38

Runs agent sandboxes on local, Docker, SSH, Daytona, Modal, or Singularity backends.

– Lists backends only, no configuration detail. product docs

55 **Checkpoints and rollback** NEW 38

Offers Checkpoints and Rollback via shadow git repos and automatic snapshots as filesystem safety nets for destructive operations.

– Names mechanism (shadow git repos) but no commands. product docs

56 **Secrets manager integrations** NEW 38

Integrates secrets from Bitwarden Secrets Manager, 1Password, and a Command Helper Secret Source.

– Names three sources, no setup steps given. product docs

57 **Scratch directory pruning warnings** NEW 38

Warns when a directory over 1 GB exists elsewhere under `cache/` with no pruner coverage, and reports the scratch directory and its size.

– Describes the warning but no remediation command. product docs

58 **Git worktrees for concurrent agents** NEW 37

Supports git worktrees for running multiple Hermes agents safely on the same repository with isolated checkouts.

— Single-sentence capability with no configuration detail. product docs

59 **New video models in the model catalog** NEW 33

Adds LTX 2.5 and Kling O3 to the video model catalogs.

— Names two models but no usage detail. v2026.9.21

60 **Fully local, account-free operation** NEW 32

Runs fully on-device with local models — no account, no API key, no data leaving the machine.

— States the property but no setup guidance. product docs

61 **Session search bounds and recall retry** NEW 32

Adds `session_search` after/before bounds and OR-relaxed recall retry.

— One-line mention with no parameters or example shown. v2026.9.21

62 **Profile Distributions for sharing agent config** NEW 30

Supports Profile Distributions for sharing a complete agent configuration as a distributable package.

— One-line description, no format or command detail. product docs

63 **Animated mascots (Pets/Petdex)** NEW 25

Adopts animated mascots (Pets / Petdex) that react to agent activity across the CLI, TUI, and desktop app.

— Cosmetic feature described in one line. product docs

v2026.9.21 NOTES ↗ CODE ↗ LATEST STABLE

Hermes v2026.9.21 adds unified connector setup across Desktop/TUI/CLI, plugin uninstall, configurable MCP discovery concurrency, and more.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2026.9.21
https://github.com/NousResearch/hermes-agent.git
# already have the repo? check out this version:
$ git checkout v2026.9.21
```

└──▷ USE IT

Throttle MCP discovery to avoid overwhelming many servers at once, or lift the cap entirely in a high-trust environment.

```yaml
mcp:
  discovery_concurrency: 8
```

Stream a query's response as newline-delimited JSON for pipeline ingestion or real-time log forwarding.

```
$ hermes --format stream-json 'summarize the last 24 h of alerts'
```

Pin a skill into every new session so it is always available without manual loading.

```yaml
skills:
  auto_load:
    - threat-intel-lookup
```

Adds `mcp.discovery_concurrency` config key (default 4, 0 = unlimited) to control how many MCP servers are contacted in parallel during discovery.

Adds `plugins.manage` remove action in the gateway and a one-click Uninstall button with confirmation on the Desktop Plugins hub.

Adds `--format stream-json` to the CLI for structured JSONL output.

Adds `skills.auto_load` config key to pin skills into every new session's prompt.

Adds `session_search` after/before bounds and OR-relaxed recall retry.

+9 more

WAS THIS USEFUL? 👍 👎

LangChain LangSmith

Feedback ⓘ announced · Sep 21, 2026

BLOG ↗ RANK

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

> LangSmith's biggest addition this window is a Trajectory/Turns view for inspecting and exporting thread conversations, alongside a new Jev-as-a-judge evaluator from TypeSafe and a shared system sandbox snapshot catalog; dozens of smaller API, Gateway, sandbox, and UI additions round out tracing, evaluation, and deployment surfaces.

└──▷ **WHAT SHIPPED · 39 FEATURES** most completely described first ? what's the number?

01 **System sandbox snapshot catalog** NEW 85

Introduces system sandbox snapshots (`system/default:latest` , `system/custom-apps:latest` , `system/code-evaluators:latest`) as a shared read-only catalog, selectable from the same sandbox and Fleet selectors as workspace snapshots. `system/code-evaluators:latest` includes NumPy, pandas, jsonschema, SciPy, and scikit-learn preinstalled so evaluators can use ACE's supported Python packages without installing them.

Create a sandbox from the prebuilt code-evaluator snapshot so your evaluators have NumPy, pandas, SciPy, and scikit-learn available without a manual install step.

> 📍 In the LangSmith console, go to Sandboxes › New Sandbox, select `system/code-evaluators:latest` from the system snapshots catalog, and click Create.

— Snapshot names, packages, and setup steps all given. changelog-20260922-f4610758

02 **Trajectory and Turns views for threads** NEW 85

Adds a Trajectory view tab on threads that renders the full conversation with human and AI turns, tool calls, and tool results in a scrollable, expand/collapse card layout, alongside a Turns view that shows each thread turn as a card with the root run's inputs and outputs for structural scanning. Supports downloading a thread as a Markdown file via the download button in the Trajectory view, including the full conversation transcript with tool calls and tool results. Adds metadata key support on individual runs to control Trajectory view visibility — runs can be excluded from the Trajectory view by importing and applying the exclusion constant.

Export a thread conversation for offline review or sharing with teammates who don't have LangSmith access.

📍 In the thread panel, open the Trajectory view, then click the download button to save the full conversation — including tool calls and tool results — as a Markdown file.

— Views, export, and exclusion constant all named with example. `product docs`

03 **Jev-as-a-judge evaluator via TypeSafe** NEW 80

Adds `TYPESAFE_API_KEY` provider secret (under Settings > Provider secrets) to connect TypeSafe AI as a model provider, and `jev-latest` as a selectable model under the TypeSafe provider in LLM-as-a-Judge evaluator configuration, supporting noul (yes/no probability), choice, and score question types. Jev-as-a-judge evaluators appear in the Evaluators tab of any tracing project, where each typed question maps to its own feedback key that can be filtered, charted, or used to trigger alerts and automations.

— Secret, model ID, and question types all named. `launch-20260921-c4267686`

04 **Thread stats endpoint** NEW 75

Adds `GET /v2/threads/stats` endpoint to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project — including on self-hosted ClickHouse-backed deployments.

— Full endpoint and returned fields named. `snapshot-20260922`

05 **Thread-trace API pagination and filtering** NEW 70

Adds a `TURN_NUMBER` field when listing thread traces to return each trace's chronological position across pages, including on public shared threads, plus new thread-trace API query parameters for filtering root runs and matching runs anywhere in a trace tree.

— Field and endpoint context named, mechanism explained. `changelog-20260922-f4610758`

06 **Configurable license expiration warning** NEW 70

Adds `LICENSE_EXPIRATION_WARNING_DAYS` environment variable so self-hosted LangSmith operators can configure how many days before license expiration the warning banner appears (default: 7).

Warn the ops team two weeks before a self-hosted LangSmith license expires instead of the default seven days.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

— Env var, default, and example config given. `snapshot-20260922`

07 **LLM Gateway provider tagging** NEW 70

Adds `ls_provider` field on OpenAI-compatible model configurations in the LLM Gateway to populate trace metadata and match custom models to provider-specific pricing.

Tag a custom OpenAI-compatible model with its provider so the LLM Gateway populates trace metadata and applies the right pricing.

```json
{
  "model": "my-custom-gpt4o",
  "base_url": "https://my-endpoint.example.com/v1",
  "ls_provider": "openai"
}
```

— Field name and example config given. snapshot-20260922

---

08 **OAuth authorization session ownership options** NEW 70

OAuth authorization sessions now accept `owner_type` and `owner_id` parameters, allowing a managed credential to be authorized for either a user or an agent, and now also accept workspace owners, enabling one managed OAuth credential to be shared across an entire workspace.

— Both new params named, mechanism explained. snapshot-20260922

---

09 **Portable Gateway prompt manifests** IMPROVED 70

Prompts saved with a LangSmith Gateway model now use a standard `ChatOpenAI` manifest with the unified Gateway URL and a `LANGSMITH_GATEWAY_API_KEY` secret reference, making them portable with standard LangChain SDKs.

— Manifest type and secret name both given. snapshot-20260922

---

10 **Dataset export permission requirement** BREAKING 65

Dataset CSV and JSONL exports now require both dataset read and download permissions; direct API requests without download permission return a forbidden response, so existing API clients with read-only permission will receive a forbidden response on export requests.

— Permission model and failure mode both named. snapshot-20260922

---

11 **Add-to-dataset and annotation queue from thread views** NEW 65

The thread header now includes Add to dataset and Add to annotation queue buttons in every thread view (Messages, Turns, and Details), so a whole thread can be added without selecting a run; the add-to-dataset popup for threads now also offers a New Dataset option that creates an empty dataset and adds the selected threads to it.

– Button and popup labels named, exact UI path. `snapshot-20260922`

12 **Evaluators list API agent filter** NEW 60

Adds `agent_id` filter to the evaluators list API to return only evaluators attached to that agent's environments or tagged datasets.

– Endpoint and filter param named, no usage example. `changelog-20260922-f4610758`

13 **Tracing batch upload timeout status codes** IMPROVED 60

`/runs/multipart` and `/runs/batch` endpoints now return `408 Request Timeout` when a tracing batch upload exceeds the server wait time, and `504` when the timeout is server-side.

– Endpoints and status codes named; behavior explained. `changelog-20260922-f4610758`

14 **Scale-to-zero rollback version guard** BREAKING 60

Scale-to-zero deployments now reject rollbacks to revisions below LangGraph API `0.13.0` or without a known compatible version; the rollback dialog directs you to rebuild the target commit with a supported version.

– Version threshold and dialog guidance both named. `changelog-20260922-f4610758`

THINNER COVERAGE BELOW

15 **Annotation queue item count time-range filters** NEW 58

Adds `min_start_time` and `max_start_time` query parameters to the annotation queue item count endpoint for time-range-scoped counts.

– Endpoint and params named, no example given. `changelog-20260922-f4610758`

16 **SDK-managed Insights report configs** NEW 55

The LangSmith Go and Java SDKs can now save, list, update, and delete Insights report configurations — including recurring schedules — enabling programmatic creation of UI-visible Insights reports.

– SDKs and capability named; no exact call given. `changelog-20260922-f4610758`

17 **Granular usage cost breakdowns** NEW 55

Adds cost estimate breakdowns for individual sandboxes and snapshots in Granular usage, with filtering by workspace, date, and resource type and CSV export.

– Feature named with filters and export format, no numbers. `changelog-20260922-f4610758`

18 **IAM-role ECR access for Sandboxes** IMPROVED 55

Commercial private ECR registries in Sandboxes can now assume an AWS IAM role on demand, enabling repository and tag search without expiring saved authorization tokens.

— Mechanism explained, no named config surface. snapshot-20260922

19 **Thread-grouped automation webhooks** NEW 50

Automations grouped by thread can now send webhook payloads containing matched conversations and their runs.

— Named trigger and payload contents; no example. changelog-20260922-f4610758

20 **Annotation queue bulk add-to-dataset** NEW 50

Adds completed annotation queue items action to `Add to Dataset`, importing each selected thread as a multi-turn example.

— Action and outcome named, no exact steps. changelog-20260922-f4610758

21 **Engine inspects tool result contents** IMPROVED 50

Engine now reads the contents of successful tool results — not only their status — to file issues where a call returned less data than claimed or a value arrived truncated.

— Mechanism explained but no named surface. changelog-20260922-f4610758

22 **LLM Gateway fallback timeout handling** IMPROVED 50

LLM Gateway fallback chains can now advance to the next model when a candidate does not return response headers within a configured timeout, with active streams continuing after headers arrive.

— Mechanism described, no named config surface. snapshot-20260922

23 **Thread query SSE streaming** NEW 50

Thread query clients can request server-sent events (SSE) to receive query results progressively while preserving cursor pagination.

— Named mechanism (SSE) but no endpoint given. snapshot-20260922

24 **Bedrock Playground timeout and retry options** IMPROVED 50

Amazon Bedrock models in the Playground now expose request timeout and max retries options, removing the previous 60-second AWS client default cutoff for long-running agent calls.

— Named options and removed default explained. snapshot-20260922

25 **Custom apps list creator filter** NEW 45

Adds filtering of the custom apps list by creator, with the selection preserved in the page URL.

– Thin one-line UI addition. changelog-20260922-f4610758

26 **Managed Deep Agents MCP OAuth discovery fallback** IMPROVED 45

Managed Deep Agents can discover OAuth settings from an MCP server's authentication challenge when well-known metadata lookup fails.

– Mechanism named but no config surface given. changelog-20260922-f4610758

27 **Traces table column customization** IMPROVED 45

The Columns menu on a tracing project's traces and runs table now allows deselecting Status and Name, and dragging them into any order.

– Exact UI path named, thin mechanism. changelog-20260922-f4610758

28 **Cache Hit % project stat** NEW 40

Adds Cache Hit % to the project stats sidebar, surfacing the share of input tokens served from the prompt cache.

– Named metric and location; no mechanism detail. changelog-20260922-f4610758

29 **Persistent LangSmith Chat conversations** IMPROVED 40

LangSmith Chat now persists your conversation as you navigate and reload within a browser tab, starting a new one in a new tab or after a browser restart with earlier conversations accessible from history.

– Behavior described, no named config surface. changelog-20260922-f4610758

30 **Dedicated trajectory batch read service** IMPROVED 40

Trajectory batch reads now run on a dedicated in-cluster gRPC service, so load from trajectory evaluations no longer competes with UI and API traffic.

– Infra change named, no reader action. snapshot-20260922

31 **MCP connector Client ID Metadata Documents** IMPROVED 40

LangSmith MCP connectors now use Client ID Metadata Documents when supported, with dynamic client registration retained as a compatibility fallback.

– Named mechanism, no config surface for reader. snapshot-20260922

32 **Dashboard feedback key suggestions** IMPROVED 35

Dashboard chart feedback key suggestions now include keys from both run-level and thread-level feedback.

– Thin description of suggestion source change. changelog-20260922-f4610758

33 **Local red-team checks against Dev targets** NEW 35

Supports running local production-parity red-team checks against supported Dev targets with their configured trace project and repository context.

— Thin description, no named surface. changelog-20260922-f4610758

34 **Slack mention-mode group DM replies** IMPROVED 35

Newly created Slack apps in mention mode respond to explicit mentions in group DMs and their threads.

— Thin behavioral description, no named surface. changelog-20260922-f4610758

35 **Claude Fable 5.1 availability in Playground** NEW 35

Claude Fable 5.1 is now available in the Anthropic, Bedrock, and Vertex AI model selectors in the Playground.

— One-line model availability note. snapshot-20260922

36 **IssueBench Opus 5 regression benchmark** NEW 35

IssueBench now includes a benchmark task that compares pre- and post-Opus 5 agent traces for cost and trajectory regressions.

— Benchmark task named, no metric detail. snapshot-20260922

37 **Workspace switcher accepts workspace IDs** IMPROVED 30

The workspace switcher now accepts workspace IDs in addition to workspace names.

— Thin one-line UI addition. snapshot-20260922

38 **Managed Deep Agent missing-connection messaging** IMPROVED 30

Managed Deep Agent channel replies now name a missing workspace connection and show how an agent developer can configure it, instead of reporting a generic run failure.

— Behavior described, no named surface. snapshot-20260922

39 **Branded OAuth connection result pages** IMPROVED 30

Agent OAuth connections now show a provider-branded result page after authorization, making successful and failed connection attempts visually distinct.

— UI change described, no named surface. snapshot-20260922

└──▷ **BREAKING ON UPGRADE**

! Scale-to-zero deployments block rollbacks to revisions below LangGraph API `0.13.0` or without a known compatible version; the rollback dialog directs you to rebuild the target commit with a supported version.

! Dataset CSV and JSONL exports now require both dataset read and download permissions; existing API clients that have read-only permission will receive a forbidden response on export requests.

announced · Sep 21, 2026 BLOG ↗

LangSmith adds TypeSafe's Jev as a judge model for online evaluations, enabling fast, low-cost typed scoring of agent traces.

Adds `TYPESAFE_API_KEY` provider secret (under Settings > Provider secrets) to connect TypeSafe AI as a model provider in LangSmith.

Adds `jev-latest` as a selectable model under the TypeSafe provider in LLM-as-a-Judge evaluator configuration, supporting noul (yes/no probability), choice, and score question types.

Supports Jev-as-a-judge evaluators in the Evaluators tab of any tracing project, where each typed question maps to its own feedback key that can be filtered, charted, or used to trigger alerts and automations.

WAS THIS USEFUL? 👍 👎

LlamaIndex

Feedback ⓘ v0.14.25 NOTES ↗ RANK

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

> LlamaIndex shipped pre-built agentic workflow classes for tool-calling LLM agents and added native async support to its LLM-based reranker.

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

01 **Pre-built agent workflow classes** NEW 80

Adds `FunctionAgent` (imported from `llama_index.core.agent.workflow`) for building function/tool-calling agents from an array of Python functions, an LLM, and a `system_prompt`. Also adds `AgentWorkflow` for orchestrating multiple agents together, plus `ReActAgent` and `CodeActAgent` as alternative pre-built agents that use different prompting strategies to execute tools.

Build a tool-calling agent from plain Python functions and run it asynchronously — useful for wiring up any callable as an LLM-driven tool.

python

```python
from llama_index.llms.openai import OpenAI
from llama_index.core.agent.workflow import FunctionAgent
import asyncio

def multiply(a: float, b: float) -> float:
    """Multiply two numbers and returns the product"""
    return a * b

def add(a: float, b: float) -> float:
    """Add two numbers and returns the sum"""
    return a + b

llm = OpenAI(model="gpt-4o-mini")
workflow = FunctionAgent(
    tools=[multiply, add],
    llm=llm,
    system_prompt="You are an agent that can perform basic
mathematical operations using tools."
)

async def main():
    response = await workflow.run(user_msg="What is 20+(2*4)?")
    print(response)

asyncio.run(main())
```

— Runnable example given but internal mechanism of ReAct/CodeAct prompting not detailed

product docs

THINNER COVERAGE BELOW

02 **Async support in StructuredLLMRerank** IMPROVED 35

Adds native async support to `StructuredLLMRerank` in `llama-index-core`.

— Names the exact class but gives no usage example or mechanism detail v0.14.25

v0.14.25 NOTES ↗ LATEST STABLE

LlamaIndex v0.14.25 adds native async support to StructuredLLMRerank in llama-index-core.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.14.25 https://github.com/run-llama/llama_index.git
# already have the repo? check out this version:
$ git checkout v0.14.25
```

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

Composio

Feedback ⓘ @composio/slim@0.19.0 NOTES ↗ RANK

Composio connects AI agents to external tools and services with managed authentication and integrations.

> Composio's SDK release focused on session reliability and safety: sessions gained a connection-guard method and exposed live server-side config, while the core client added typed tool-fetch errors, an SSRF strict mode, and a documented upload size cap; a separate slim release added error-handling for Pusher-based trigger subscriptions.

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

01 session.ensureConnected() guards against duplicate auth flows

NEW

85

session.ensureConnected(toolkit, options?) returns immediately if the session already has an active connection for the toolkit (or the toolkit is no-auth); otherwise it starts the authorization flow via session.authorize() and waits for the new connection to become active, preventing duplicate pending connections in long-running agent sessions.

Ensure a toolkit is connected before executing a tool, without risking duplicate pending auth flows in a long-running agent session.

```text
await session.ensureConnected('github');
const result = await session.execute('GITHUB_LIST_REPOS', {});
```

— Named method and params with a runnable usage example. @composio/core@0.19.0

02 Live session config exposed via session.config

BREAKING

78

Sessions now expose server-side configuration through `session.config`, carrying toolkit/tool allowlists, tags, auth configs, connected accounts, `manage_connections`, preload and sandbox settings as returned by create(), use(), and update(). This is now a required member of the `Session` interface, so test doubles or wrappers built to satisfy `Session` must include a `config` value or TypeScript compilation will fail.

Inspect the live server-side allowlist and settings for a session immediately after updating it, without dropping to the raw API client.

```text
const config = await session.update({ toolkits: ['github',
'slack'] });
console.log(config.toolkits.enabled, config.manage_connections);
```

— Names the config surface and flags the interface break precisely. `@composio/core@0.19.0`

03 onSubscriptionError callback for Pusher and trigger subscriptions

NEW 65

Adds an optional `onSubscriptionError` callback parameter to `PusherService.subscribe` and `Triggers.subscribe` so applications can react programmatically to Pusher subscription failures instead of surfacing them as unhandled promise rejections.

Handle Pusher subscription failures in a trigger listener so errors don't surface as unhandled promise rejections.

```text
await Triggers.subscribe('MY_TRIGGER', handler, {
  onSubscriptionError: (err) => console.error('Subscription
failed:', err)
});
```

— Named parameter and methods with a runnable example. `@composio/slim@0.19.0` `@composio/core@0.19.0`

THINNER COVERAGE BELOW

04 ComposioToolFetchError distinguishes fetch failures from not-found

NEW 55

Introduces `ComposioToolFetchError` (preserving the original client error as `cause`) for non-404/400 failures in `tools.getRawComposioToolBySlug`, `tools.get`, and `tools.execute`, separating network/auth failures from genuine not-found errors.

— Named error type and methods but no usage example. `@composio/core@0.19.0`

05 Exported upload size cap for URL downloads NEW 50

Exports `readResponseBodyWithLimit` and `MAX_URL_UPLOAD_SIZE_BYTES` so downstream packages can apply the SDK's 100 MiB cap when downloading files from user-supplied URLs.

— Exact export names and limit given, no usage shown. `@composio/core@0.19.0`

06 **Strict mode for ssrfSafeFetch** NEW 45

Adds an opt-in strict mode to `ssrfSafeFetch` that rejects configured network routes when the connection cannot be pinned to the validated address.

— Names the function and behavior but no configuration example. `@composio/core@0.19.0`

└──▷ **BREAKING ON UPGRADE**

! `config` is now a required member of the `Session` interface — test doubles, wrappers, or any object you construct to satisfy `Session` must include a `config` value or TypeScript compilation will fail.

`@composio/slim@0.19.0` NOTES ↗ LATEST STABLE

Composio slim 0.19.0 adds an `onSubscriptionError` callback to `PusherService.subscribe` and `Triggers.subscribe` for programmatic error handling.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @composio/slim@0.19.0
https://github.com/ComposioHQ/composio.git
# already have the repo? check out this version:
$ git checkout @composio/slim@0.19.0
```

Adds an optional `onSubscriptionError` callback parameter to `PusherService.subscribe` and `Triggers.subscribe` so applications can react programmatically to Pusher subscription failures.

`@composio/core@0.19.0` NOTES ↗ STABLE

Composio @composio/core@0.19.0 adds session.ensureConnected(), exposes live session config, and surfaces new error types for tool fetch failures.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @composio/core@0.19.0
https://github.com/ComposioHQ/composio.git
# already have the repo? check out this version:
$ git checkout @composio/core@0.19.0
```

└──▷ **USE IT**

Ensure a toolkit is connected before executing a tool, without risking duplicate pending auth flows in a long-running agent session.

```text
await session.ensureConnected('github');
const result = await session.execute('GITHUB_LIST_REPOS', {});
```

Inspect the live server-side allowlist and settings for a session immediately after updating it, without dropping to the raw API client.

```text
const config = await session.update({ toolkits: ['github', 'slack']
});
console.log(config.toolkits.enabled, config.manage_connections);
```

Handle Pusher subscription failures in a trigger listener so errors don't surface as unhandled promise rejections.

```text
await Triggers.subscribe('MY_TRIGGER', handler, {
  onSubscriptionError: (err) => console.error('Subscription
failed:', err)
});
```

Adds session.ensureConnected(toolkit, options?) that returns immediately if the session already has an active connection for the toolkit (or the toolkit is no-auth), and otherwise starts the authorization flow via session.authorize() and waits for the new connection to become active — preventing duplicate pending connections.

Exposes server-side session configuration on sessions via `session.config`, carrying toolkit/tool allowlists, tags, auth configs, connected accounts, `manage_connections`, preload and sandbox settings returned by create(), use(), and update().

Introduces `ComposioToolFetchError` (with the client error preserved as `cause`) for non-404/400 failures in `tools.getRawComposioToolBySlug`, `tools.get`, and `tools.execute`, separating network/auth failures from genuine not-found errors.

Exports `readResponseBodyWithLimit` and `MAX_URL_UPLOAD_SIZE_BYTES` so downstream packages can apply the SDK's 100 MiB cap when downloading files from user-supplied URLs.

Adds an opt-in strict mode to `ssrfSafeFetch` that rejects configured network routes when the connection cannot be pinned to the validated address.

└──▷ BREAKING ON UPGRADE

! `config` is now a required member of the `Session` interface — test doubles, wrappers, or any object you construct to satisfy `Session` must include a `config`

value or TypeScript compilation will fail.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Snowflake Cortex AI

Feedback ⓘ changelog · Sep 22, 2026

CHANGELOG ↗ RANK

Snowflake Cortex AI provides managed language-model functions, search, and agents that work with data governed in Snowflake.

> Snowflake shipped a broad release headlined by Cortex AI Gateway-adjacent additions to Cortex Agents (a new Compact API, object enhancements, and Native Apps code execution tools), new Cortex AI Function Evaluation and Optimization tools, and an `EXPLAIN CHANGES` command for previewing dynamic table refresh impact, alongside catalog, Iceberg, Streamlit, container services and quota-control improvements.

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

01 **EXPLAIN CHANGES for dynamic tables** NEW 75

Adds `EXPLAIN CHANGES` (now GA) to predict dynamic table refresh behavior before changes are applied, letting users preview cost and scope impact of a schema change before committing it, e.g. `EXPLAIN CHANGES ON DYNAMIC TABLE my_db.my_schema.my_dynamic_table;`.

Preview how a schema change will affect dynamic table refresh costs and scope before committing it.

```
$ EXPLAIN CHANGES ON DYNAMIC TABLE
my_db.my_schema.my_dynamic_table;
```

— Named command with a runnable example, but limited mechanism detail.

changelog-20260922-ec2c804d

02 **Per-user quota controls** NEW 60

Adds per-user quota controls that let admins include or exclude specific users by name and set weekly per-user quota limits, now GA.

— Describes concrete mechanism (by name, weekly limits) but no exact config path.

changelog-20260922-ec2c804d

THINNER COVERAGE BELOW

03 **Horizon Catalog Explorer UI** NEW 45

Launches the Horizon Catalog Explorer UI for browsing and managing the Snowflake catalog, currently in Preview.

— Names a UI a reader can navigate to, but no path or workflow given.

changelog-20260922-ec2c804d

04 **Streamlit embedding within external pages** NEW 40

Enables embedding Streamlit in Snowflake apps within external pages, now GA.

— Names the surface but no path or config detail given. changelog-20260922-ec2c804d

05 **Partition evolution for Apache Iceberg tables** NEW 40

Adds partition evolution support for Apache Iceberg™ tables, now GA.

— Names Iceberg and GA status but no mechanism or example. changelog-20260922-ec2c804d

06 **Cortex Agents Compact API** NEW 35

Introduces the Cortex Agents Compact API, a leaner API surface for Cortex Agents interactions, currently in Preview.

— Names the API but gives no endpoint, method, or usage detail.

changelog-20260922-ec2c804d

07 **Cortex AI Function Evaluation and Optimization** NEW 35

Introduces Cortex AI Function Evaluation for measuring the quality of AI function outputs and Cortex AI Function Optimization for producing more efficient AI implementations, both in Public Preview.

— Two related previews described only in one line each, no mechanism given.

changelog-20260922-ec2c804d

08 **Write support for nested namespaces in catalog-linked databases** NEW 35

Adds write support for nested namespaces in catalog-linked databases.

— States the capability but no mechanism or example. changelog-20260922-ec2c804d

09 **Snowflake Native Apps code execution tools for Cortex Agents** NEW 35

Adds Snowflake Native Apps code execution tools for use with Cortex Agents.

— One-line description with no usage or configuration detail. changelog-20260922-ec2c804d

10 **Instance type selection for Snowpark Container Services backups** IMPROVED 35

Supports Snowpark Container Services backup instance types, allowing selection of instance types for container backups, now GA.

— Names the surface but no config key or example given. changelog-20260922-ec2c804d

11 **Cortex Agents object enhancements** IMPROVED 25

Adds Cortex Agents object enhancements, expanding the set of configurable agent objects, now GA.

— Vague description with no named objects or fields. changelog-20260922-ec2c804d

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Cortex AI Gateway, Compact API, Function Evaluation/Optimization, and EXPLAIN CHANGES for dynamic tables headline this Snowflake release.

└──▷ TRY IT

Preview how a schema change will affect dynamic table refresh costs and scope before committing it.

```
$ EXPLAIN CHANGES ON DYNAMIC TABLE
my_db.my_schema.my_dynamic_table;
```

Adds `EXPLAIN CHANGES` to predict dynamic table refresh behavior before changes are applied (GA).

Introduces Cortex Agents Compact API, a leaner API surface for Cortex Agents interactions (Preview).

Introduces Cortex AI Function Evaluation for measuring the quality of AI function outputs (Public Preview).

Introduces Cortex AI Function Optimization for producing more efficient AI implementations (Public Preview).

Adds write support for nested namespaces in catalog-linked databases.

+7 more

WAS THIS USEFUL? 👍 👎

fal

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗ RANK

fal provides hosted generative-media model APIs and a serverless GPU platform for deploying custom inference workloads.

> fal added billing headers for WebSocket endpoints to support usage-based billing on streaming connections, and introduced a Playground for testing any endpoint of a deployed Serverless app directly from the dashboard.

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

01 **Billing headers for WebSocket endpoints** NEW 85

WebSocket upgrades now support `x-fal-billable-units` to declare a known unit count at connect time, and `x-fal-billable-units-webhook` to defer reporting until session end; totals are then submitted via `POST /requests/billable-units/{request_id}`.

– Names exact headers and endpoint but no full request/response example.

changelog-20260922-f40bb0e0

THINNER COVERAGE BELOW

02 **Playground for testing Serverless app endpoints** NEW 53

A new `Testing > Playground` section in the app dashboard for deployed Serverless apps lets users select any endpoint, fill its auto-generated input form, and run requests without writing a client.

Test all endpoints of a multi-endpoint Serverless app in one place without writing a client.

📍 1. Open your app in the fal dashboard. 2. Navigate to Testing › Playground. 3. Select the endpoint you want to test from the dropdown. 4. Fill in the generated input form and click Run.

– Clear UI navigation path but no code or config surface. changelog-20260922-f40bb0e0

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

fal adds billing headers for WebSocket endpoints and a multi-endpoint Playground in the app dashboard.

└──▷ HOW TO FIND IT

Test all endpoints of a multi-endpoint Serverless app in one place without writing a client.

📍 1. Open your app in the fal dashboard. 2. Navigate to Testing › Playground. 3. Select the endpoint you want to test from the dropdown. 4. Fill in the generated input form and click Run.

Adds `x-fal-billable-units` response header on WebSocket upgrade to declare a known unit count at connect time, and `x-fal-billable-units-webhook` to defer reporting to session end, with totals submitted via `POST /requests/billable-units/{request_id}`.

Adds `Testing > Playground` section to the app dashboard for deployed Serverless apps, letting you select any endpoint, fill its generated input form, and run requests without leaving the app.

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock

Feedback

changelog · Sep 22, 2026

CHANGELOG ↗ RANK

Amazon Bedrock provides managed foundation-model inference, customization, knowledge bases, and APIs for generative AI applications.

> Amazon Bedrock's documentation now steers OpenAI-compatible API users toward the `bedrock-runtime` endpoint and clarifies how to discover models without the `/models` operation.

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

01 **OpenAI-compatible API guidance on bedrock-runtime** IMPROVED

73

Bedrock now recommends `bedrock-runtime` as the endpoint for new Responses API and Chat Completions API integrations, replacing the previous `-named` path. Because the OpenAI-compatible `/models` operation is not available on `bedrock-runtime`, docs direct users to `ListFoundationModels`, `ListInferenceProfiles`, and model cards for model discovery instead, and the Quickstart has been updated to use a system inference profile that supports both OpenAI-compatible APIs on `bedrock-runtime`.

— Names exact endpoints and API calls but no code samples given

changelog-20260922-f94fa30e

---

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Amazon Bedrock's OpenAI-compatible APIs now prefer bedrock-runtime, with guidance on model discovery via ListFoundationModels and ListInferenceProfiles.

Establishes `bedrock-runtime` as the recommended endpoint for new Responses API and Chat Completions API integrations, replacing the previous `-named` path.

Documents that the OpenAI-compatible `/models` operation is not available on `bedrock-runtime`, directing users to `ListFoundationModels`, `ListInferenceProfiles`, and model cards for model discovery instead.

Updates the Quickstart to use a system inference profile that supports both OpenAI-compatible APIs on `bedrock-runtime`.

WAS THIS USEFUL? 👍 👎

RunPod

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗ RANK

AI infrastructure with on-demand GPUs and serverless compute. Run training, inference, and batch workloads on the cloud with Runpod.

> RunPod introduced Global Volumes (Beta) for region-independent Pod storage, added business tax ID handling for credit purchases, and is retiring the Nano Banana Edit endpoint in favor of Nano Banana 2 Edit.

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

01 **Nano Banana Edit endpoint retirement** DEPRECATED 60

The Nano Banana Edit public endpoint will be retired on September 28, 2026; users must migrate to Nano Banana 2 Edit before that date.

– Gives exact date and required migration target changelog-20260922-1a834be0

THINNER COVERAGE BELOW

02 **Global Volumes (Beta) for Pods** NEW 50

Adds elastic, region-independent storage that can be attached to any Pod across any data center, removing the need to tie storage to a single region.

– Names the feature and behaviour but no setup steps or limits

changelog-20260922-1a834be0

03 **Business tax ID entry for credit purchases** NEW 50

Adds business tax ID entry at checkout and in account settings to control how sales tax applies to credit purchases.

– Names exact UI locations but no further mechanism detail changelog-20260922-1a834be0

└──▷ BREAKING ON UPGRADE

! The Nano Banana Edit public endpoint will be retired on September 28, 2026; migrate to Nano Banana 2 Edit before that date.

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

RunPod adds Global Volumes (Beta) — elastic, region-independent storage attachable to any Pod across any data center.

Adds business tax ID entry at checkout and in account settings to control how sales tax applies to credit purchases.

└──▷ BREAKING ON UPGRADE

! The Nano Banana Edit public endpoint will be retired on September 28, 2026; migrate to Nano Banana 2 Edit before that date.

WAS THIS USEFUL? 👍 👎

Fireworks AI

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗

RANK

Fireworks AI provides hosted inference, model fine-tuning, and deployment APIs for open-weight models.

> Fireworks AI added deployment-shape controls to its Create Deployment API, firectl, and Python SDK — one flag to auto-select a validated shape and another to explicitly opt out, with opt-out enforcement coming soon.

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

01 Explicit opt-out for shapeless deployments ahead of enforcement

95

BREAKING

Adds `acceptShapelessRisk: true` to the Create Deployment API, `--accept-shapeless-risk` in firectl, and `accept_shapeless_risk=True` in the Python SDK as an explicit opt-out to create a deployment without a shape, preserving current behavior for advanced users. Shapeless deployment creation will soon require passing this flag explicitly — deployments that currently omit a shape without it will fail once enforcement is enabled.

Preserve shapeless deployment behavior for a workload with non-standard requirements while enforcement is pending.

```
$ firectl create deployment --model
accounts/fireworks/models/deepseek-v4-1-flash --accept-
shapeless-risk
```

— Names flags, gives before/after and future enforcement timeline.

changelog-20260922-35f061d0

02 Auto-selected deployment shape via `deploymentShape` flag NEW

90

Adds `deploymentShape: "default"` to the Create Deployment API, `--deployment-shape default` in firectl, and `deployment_shape="default"` in the Python SDK, letting Fireworks auto-select a validated shape for a model instead of requiring manual configuration. This is intended to avoid the most common cause of failed deployments.

Let Fireworks pick a validated deployment shape automatically to avoid the most common cause of failed deployments.

```
$ firectl create deployment --model
accounts/fireworks/models/deepseek-v4-1-flash --deployment-shape
default
```

— Names exact flags across API, CLI and SDK with runnable command.

changelog-20260922-35f061d0

└──▷ **BREAKING ON UPGRADE**

! Shapeless deployment creation will soon require passing `acceptShapelessRisk=true` (API/SDK) or `--accept-shapeless-risk` (firectl) explicitly — deployments that currently omit a shape without this flag will fail once enforcement is enabled.

changelog · Sep 22, 2026 | CHANGELOG ↗ | LATEST | STABLE

Fireworks AI adds `deploymentShape` and `acceptShapelessRisk` flags to the Create Deployment API, firectl, and Python SDK.

└──▷ **TRY IT**

Let Fireworks pick a validated deployment shape automatically to avoid the most common cause of failed deployments.

```
$ firectl create deployment --model
accounts/fireworks/models/deepseek-v4-1-flash --deployment-shape
default
```

Preserve shapeless deployment behavior for a workload with non-standard requirements while enforcement is pending.

```
$ firectl create deployment --model
accounts/fireworks/models/deepseek-v4-1-flash --accept-shapeless-
risk
```

Adds `deploymentShape: "default"` to the Create Deployment API, `--deployment-shape default` in firectl, and `deployment_shape="default"` in the Python SDK, letting Fireworks auto-select a validated shape for your model instead of requiring manual configuration.

Adds `acceptShapelessRisk: true` to the Create Deployment API, `--accept-shapeless-risk` in firectl, and `accept_shapeless_risk=True` in the Python SDK as an explicit opt-out to create a deployment without a shape, preserving current behavior for advanced users.

> **└──▷ BREAKING ON UPGRADE**
>
> **!** Shapeless deployment creation will soon require passing `acceptShapelessRisk=true` (API/SDK) or `--accept-shapeless-risk` (firectl) explicitly — deployments that currently omit a shape without this flag will fail once enforcement is enabled.

WAS THIS USEFUL?  

HeyGen HyperFrames

Feedback ⓘ v0.8.60 NOTES ↗ RANK

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

> HyperFrames shipped new CLI subcommands for agent-driven media search and timeline editing, expanded audio metering and editing controls in Studio, and added a handful of smaller Studio UI and registry updates.

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

01 CLI subcommands for agent workflows NEW

75

Adds `media-use` subcommand to the CLI for searching media assets, and `timeline workflows` subcommand to let agents edit timelines programmatically.

– Names exact CLI subcommands but no usage examples or flags given v0.8.59

02 Audio metering and editing controls in Studio IMPROVED

65

Adds audio level meters displayed at the timeline's right edge, meter faders for groups and master, and draws the audio meters toggle as a visual level meter. Also adds the ability to type exact audio values, fade clip edges, and preserve audio when dropping a video with audio.

– Concrete UI behaviors described but no exact controls or shortcuts named v0.8.60 v0.8.59

THINNER COVERAGE BELOW

03 Timeline composition from host-app-defined parts NEW

35

Adds timeline composition from host-app-defined parts in Studio.

– Thin description with no mechanism or interface named v0.8.59

04 Studio UI rendering updates IMPROVED

35

Paints menus and popovers instantly in Studio, eliminating render delay, and updates timeline clips to use the designed material.

– Generic UI polish without named surfaces or metrics v0.8.60

05 Weekly merges leaderboard block NEW

30

Adds a week in merges leaderboard block to the registry.

– Bare feature name with no further mechanism described v0.8.60

v0.8.60 NOTES ↗ LATEST STABLE

HyperFrames v0.8.60 adds audio meter faders for groups and master, a leaderboard block, and instant-painting menus in Studio.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.60 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.60
```

Draws the audio meters toggle as a visual level meter in Studio.

Adds a week in merges leaderboard block to the registry.

Paints menus and popovers instantly in Studio, eliminating render delay.

Updates timeline clips to use the designed material in Studio.

v0.8.59 NOTES ↗ STABLE

HyperFrames v0.8.59 adds CLI media search via `media-use`, agent timeline editing via `timeline workflows`, and audio meters in Studio.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.59 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.59
```

Adds `media-use` subcommand to the CLI for searching media assets.

Adds `timeline workflows` to the CLI to let agents edit timelines programmatically.

Adds audio level meters displayed at the timeline's right edge in Studio.

Adds ability to type exact audio values, fade clip edges, and preserve audio when dropping a video with audio in Studio.

Adds timeline composition from host-app-defined parts in Studio.

WAS THIS USEFUL? 👍 👎

Google Gen AI JavaScript SDK

Feedback ⓘ v2.24.0

NOTES ↗ RANK

The Google Gen AI JavaScript SDK provides TypeScript and JavaScript interfaces to Gemini APIs and live model sessions.

> Google Gen AI JavaScript SDK v2.24.0 introduces a Voices API resource and expands public speech-related types for richer introspection of voice and speech responses.

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

01 **Voices API resource and speech type exposure** NEW 71

The SDK adds a `Voices` API resource enabling programmatic listing and retrieval of available voices, adds a `sample_audio` field to the `Voice` type so callers can retrieve audio samples for a voice, and exposes `SpeechMetadata`, `VoiceConfig.voice`, and `SpeechAnnotation` in the public SDK surface for richer speech response introspection.

— Names all new types and the API resource but no usage example v2.24.0

v2.24.0 NOTES ↗ LATEST STABLE

v2.24.0 adds a Voices API resource, `sample_audio` on Voice, and exposes `SpeechMetadata`, `VoiceConfig.voice`, and `SpeechAnnotation` publicly.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.24.0 https://github.com/googleapis/js-genai.git
# already have the repo? check out this version:
$ git checkout v2.24.0
```

Exposes `SpeechMetadata`, `VoiceConfig.voice`, and `SpeechAnnotation` in the public SDK surface for richer speech response introspection.

Adds `sample_audio` field to the `Voice` type, allowing callers to retrieve audio samples for a voice.

Adds a `Voices` API resource to the SDK, enabling programmatic listing and retrieval of available voices.

WAS THIS USEFUL? 👍 👎

Google Gen AI Python SDK

Feedback ⓘ v2.25.0 NOTES ↗

RANK

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

> Google Gen AI Python SDK v2.25.0 introduces a Voices API resource along with new speech-related types for retrieving voice samples and metadata.

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

01 **Voices API resource and speech types** NEW 60

Adds a `Voices` API resource to the SDK, exposing programmatic access to voice listings and management. Adds a `sample_audio` field to the `Voice` type, enabling retrieval of audio samples for available voices. Also exposes `SpeechMetadata`, `VoiceConfig.voice`, and `SpeechAnnotation` as public types in the GenAI SDK.

— Names exact types/fields but no usage example shown. v2.25.0

v2.25.0 NOTES ↗ LATEST STABLE

Google Gen AI Python SDK v2.25.0 adds a Voices API resource, sample audio on Voice, and exposes SpeechMetadata and SpeechAnnotation.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.25.0 https://github.com/googleapis/python-genai.git
# already have the repo? check out this version:
$ git checkout v2.25.0
```

Adds `sample_audio` field to the `Voice` type, enabling retrieval of audio samples for available voices.

Adds `Voices` API resource to the SDK, exposing programmatic access to voice listings and management.

Exposes `SpeechMetadata`, `VoiceConfig.voice`, and `SpeechAnnotation` as public types in the GenAI SDK.

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.17.0 NOTES ↗ RANK

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

> OpenAI Python SDK v3.17.0 rounds out several API surfaces with new configuration and event types spanning storage, safety, sessions, SIP calls, and authentication.

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

01 **Safety warning and deactivation webhooks** NEW 35

Adds safety warning and deactivation webhook events to the API, allowing applications to receive notifications for these safety-related occurrences.

– Names the event type but no endpoint or payload details v3.17.0

02 **SIP media security fields for incoming calls** NEW 35

Adds SIP media security fields to incoming call events, surfacing security-related metadata for SIP media.

– Names the feature area but not specific field names v3.17.0

03 **Environment variable vault credentials for auth** NEW 30

Supports environment variable vault credentials for API authentication, adding a new way to supply credentials to the client.

– Mentions env-var mechanism but no exact variable name v3.17.0

04 **External storage configuration management** NEW 25

Adds external storage configuration management to the API, letting external storage settings be configured through the SDK.

– No named endpoint, flag, or config key given v3.17.0

05 **Session environment reset events** NEW 25

Adds session environment reset events to the API, exposing when a session's environment has been reset.

– Bare description, no mechanism or usage shown v3.17.0

v3.17.0 NOTES ↗ LATEST STABLE

OpenAI Python SDK v3.17.0 adds external storage config management, safety case retrieval, safety webhooks, session reset events, SIP media security, and vault

credentials.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.17.0 https://github.com/openai/openai-python.git
# already have the repo? check out this version:
$ git checkout v3.17.0
```

Adds external storage configuration management to the API.

Adds safety warning and deactivation webhook events to the API.

Adds session environment reset events to the API.

Adds SIP media security fields to incoming call events.

Supports environment variable vault credentials for API authentication.

WAS THIS USEFUL?  

OpenAI TypeScript SDK

Feedback ⓘ v7.21.0 NOTES ↗ RANK

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

> OpenAI TypeScript SDK v7.21.0 adds session environment reset events to the Realtime API.

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

01 **Session environment reset events in Realtime API** NEW

33

The SDK now supports session environment reset events, enabling clients to detect and handle environment resets within a Realtime session.

— Names the event type but no field, method, or usage example. v7.21.0

v7.21.0 NOTES ↗ LATEST STABLE

OpenAI TypeScript SDK v7.21.0 adds session environment reset events to the Realtime API surface.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v7.21.0 https://github.com/openai/openai-node.git
# already have the repo? check out this version:
$ git checkout v7.21.0
```

WAS THIS USEFUL? 👍 👎

FlashInfer

Feedback ⓘ v0.7.0 NOTES ↗ RANK

FlashInfer provides GPU kernels for language-model inference, including attention, sampling, and quantized operations.

> FlashInfer's v0.7.0 window ships new paged Multi-head Latent Attention (MLA) decode primitives alongside a large wave of Blackwell/SM100-SM120 kernels spanning attention, MoE, recurrent state-space (KDA/GDN/Mamba), and GEMM/quantization/communication backends.

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

01 **Paged MLA decode primitives** NEW 95

FlashInfer adds `prims_ts_batch_mla_decode_with_kv_cache` for fixed and packed-query paged Multi-head Latent Attention (MLA) decode with caller-owned scratch buffers, supporting packed-query mode via `qo_indptr` and `max_seq_len_q` for variable per-request query lengths in a compact `[total_q, H, 576]` tensor alongside the fixed-shape `[B, SQ, H, 576]` mode, and accepting KV cache as compact rank-3 or rank-4 `[pages, page_size, 576]` tensors with bottom-right aligned causal masking. `flashinfer.attention.prims_ts.batch_mla_decode_with_paged_kv_cache` wraps this as a one-shot convenience call supporting `mask_type` of `'dense'` or `'causal'`, a caller-supplied output tensor via `out`, and configurable `out_dtype` defaulting to `torch.bfloat16`.

Run paged MLA decode over a fixed-shape batch (all requests have the same Q length) using causal masking and a pre-allocated output buffer for CUDA graph compatibility.

```python
import torch
from flashinfer.attention.prims_ts import
prims_ts_batch_mla_decode_with_kv_cache

B, SQ, H = 4, 1, 16
kv_lora_rank, qk_rope_head_dim = 512, 64
page_size, max_pages = 16, 128
max_seq_len = 2048

query = torch.randn(B, SQ, H, 576, dtype=torch.bfloat16,
device='cuda')
kv_cache = torch.randn(max_pages, page_size, 576,
dtype=torch.bfloat16, device='cuda')
workspace_buffer = torch.empty(64 * 1024 * 1024,
dtype=torch.uint8, device='cuda')
block_tables = torch.zeros(B, max_pages // B, dtype=torch.int32,
device='cuda')
seq_lens = torch.tensor([128, 256, 512, 1024],
dtype=torch.int32, device='cuda')
out = torch.empty(B, SQ, H, kv_lora_rank, dtype=torch.bfloat16,
device='cuda')

result = prims_ts_batch_mla_decode_with_kv_cache(
    query, kv_cache, workspace_buffer,
    kv_lora_rank=kv_lora_rank,
    qk_rope_head_dim=qk_rope_head_dim,
    block_tables=block_tables,
    seq_lens=seq_lens,
    max_seq_len=max_seq_len,
    out=out,
    mask_type='causal',
    out_dtype=torch.bfloat16,
)
```

Run paged MLA decode in packed-query mode to handle variable per-request Q lengths in a single fused kernel call.

```python
import torch
from flashinfer.attention.prims_ts import
prims_ts_batch_mla_decode_with_kv_cache

B, H = 3, 16
kv_lora_rank, qk_rope_head_dim = 512, 64
page_size, max_pages = 16, 64
max_seq_len = 1024
max_seq_len_q = 8

# Packed query: requests have 2, 4, 2 query tokens respectively
qo_indptr = torch.tensor([0, 2, 6, 8], dtype=torch.int32,
device='cuda')
total_q = 8
query = torch.randn(total_q, H, 576, dtype=torch.bfloat16,
device='cuda')
kv_cache = torch.randn(max_pages, page_size, 576,
dtype=torch.bfloat16, device='cuda')
workspace_buffer = torch.empty(64 * 1024 * 1024,
dtype=torch.uint8, device='cuda')
block_tables = torch.zeros(B, max_pages // B, dtype=torch.int32,
device='cuda')
seq_lens = torch.tensor([64, 128, 256], dtype=torch.int32,
device='cuda')

result = prims_ts_batch_mla_decode_with_kv_cache(
    query, kv_cache, workspace_buffer,
    kv_lora_rank=kv_lora_rank,
    qk_rope_head_dim=qk_rope_head_dim,
    block_tables=block_tables,
    seq_lens=seq_lens,
    max_seq_len=max_seq_len,
    qo_indptr=qo_indptr,
    max_seq_len_q=max_seq_len_q,
    mask_type='causal',
    out_dtype=torch.bfloat16,
)
```

Run a single MLA decode step over a paged KV cache outside of CUDA Graph capture, using causal masking and packed queries.

```python
import torch
from flashinfer.attention.prims_ts import
batch_mla_decode_with_paged_kv_cache

output = batch_mla_decode_with_paged_kv_cache(
    query=query,          # [num_tokens, kv_lora_rank +
qk_rope_head_dim]
    kv_cache=kv_cache,    # compact paged latent K/V cache
    block_tables=block_tables,
    seq_lens=seq_lens,
    qo_indptr=qo_indptr,  # packed-query mode: cumulative query
offsets
    kv_lora_rank=512,
    qk_rope_head_dim=64,
    mask_type='causal',
    out_dtype=torch.bfloat16,
)
```

— Full function signatures, tensor shapes and runnable examples given. product docs

02 **Blackwell attention kernel additions** NEW 60

FlashInfer adds a `variant_owns_mask` flag for JIT attention variants that own the full attention mask; an `is_causal` parameter plus a prims-ts backend for paged decode attention; a native Blackwell DCP speculative decode path in the Cake FMHA backend; paged MQA logits (attn_scores) kernels for Blackwell SM100; optimized SM100/SM103 block-sparse attention (VSA) in the Cake backend; Blackwell minimax sparse attention (MSA) source kernels in the Cake backend; a PrimTS Q64/KV256 tile config with paged GQA block-sparse attention; variable-window FMHA context support for PrimTS; variable-Q decode with DCP in the CuTeDSL MLA backend; and trtllm-gen FMHA features for SM107 (spcompress and fp16softmax).

— Many named kernels/flags listed, no usage examples given. v0.7.0

03 **Blackwell MoE kernels and quantization** NEW 60

FlashInfer adds `SiTU-GLU` activation to the CUTLASS fused-MoE backend; an SM100 BF16 CuTeDSL MegaMoE expert-parallel kernel; an SM100 W4A8 (MXFP8xMXFP4) CuTeDSL split-kernel backend with MXFP8 packed dispatch for expert-parallel MoE; a deterministic prepared Blackwell MoE backend (`cake_bgmv`); FP8 per-channel quantization support in the MoE fused kernel; per-token NVFP4 ReLU2 MoE support; a TRTLLM Gen MoE split-K path; a standalone trtllm-gen routing op with decomposed MoE routing matrix tests; `B12xMoEWrapper` workspace sharing for pre-allocated MoE workspaces; the remaining CUTLASS unified MoE runners; and unified activation parity for the unified MoE backend.

— Many named MoE backends listed without usage examples. v0.7.0

04 **Recurrent state-space kernels: KDA, GDN, Mamba** NEW

60

FlashInfer adds an SM120a CuTe DSL KDA prefill backend and a CuTe DSL recurrent KDA prefill backend; paired recurrent training for SM100a and SM103a in the Cake KDA backend with recurrent training template dispatch; native unbounded-softplus Kimi-Linear kernels in the Cake KDA backend; a Qwen fused GDN decode step for SM120 and `fused_GDN_step` support for Qwen 3.6 35B A3B on SM120; pooled state, state checkpointing, and dtype support added to the GDN backend; and a Blackwell SSDCombined Mamba kernel plus a Blackwell selective state-update backend in the Cake Mamba backend.

— Many named backends listed without usage examples. v0.7.0

05 **Blackwell GEMM, quantization and communication kernels** NEW

60

FlashInfer adds a custom all-reduce kernel; a Cake Blackwell all-gather matmul communication backend; CuTe DSL kernels for Rubin (SM107) and batched FP8 GEMM for Blackwell; SM120 NVFP4 SVDQuant GEMM in CuTe DSL; bias support added to the cuBLAS GEMM backend and as a fallback for the CuTe DSL GEMM backend; a fused `Add+RMSNorm` kernel with 1x128 FP8 block-quantization; fused Blackwell SwiGLU MXFP8 quantization in the Cake activation backend; grouped MXFP8 quantization in the Cake backend; a CUB `DeviceBatchedTopK` top-k backend with variable-length sequence support; and multi-arch cubins (SM100, SM103, SM107) packaged in a single artifact.

— Many named kernels/backends listed without usage examples. v0.7.0

v0.7.0 NOTES ↗ LATEST STABLE

FlashInfer v0.7.0 adds Blackwell/SM107 kernels, CuTe DSL MoE and KDA backends, custom all-reduce, FP8 per-channel MoE quant, and more.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.7.0 https://github.com/flashinfer-ai/flashinfer.git
# already have the repo? check out this version:
$ git checkout v0.7.0
```

Adds `variant_owns_mask` flag for JIT attention variants that own the full attention mask.

Adds `is_causal` parameter and a prims-ts backend to paged decode attention.

> Adds `SiTU-GLU` activation to the CUTLASS fused-MoE backend.
>
> Adds CUB `DeviceBatchedTopK` top-k backend with variable-length sequence support.
>
> Adds fused `Add+RMSNorm` with 1x128 FP8 block-quantization producer kernel.
>
> +36 more

WAS THIS USEFUL? 👍 👎

NVIDIA Dynamo

Feedback ⓘ v1.5.0 NOTES ↗ RANK

Dynamo is a datacenter-scale distributed inference serving framework with disaggregated prefill and decode, KV-cache-aware routing, and multi-engine support.

> Dynamo v1.5.0 ships pluggable router and tokenizer policies, TLS across all transports, a Triton backend for non-LLM models, and an experimental standalone Sidecar for stock inference engines, alongside a cluster of breaking changes — a new CRD storage version, a Rust-based Endpoint Picker, a standalone Snapshot operator, and AISimulate replacing AIConfigurator.

└──▷ **WHAT SHIPPED · 26 FEATURES** most completely described first ? what's the number?

01 AISimulate replaces AIConfigurator and dynamo.replay

BREAKING

80

Replaces AIConfigurator with AISimulate (shipped as its own package at v0.12.0 with one CLI), replacing the aiconfigurator wheels and the `dynamo.replay` entrypoint; offline replay now returns a single `ReplayReport` with per-request capture opt-in.

— Names package, CLI, and replaced entrypoint with version. v1.5.0

02 Dynamo Snapshot promoted to standalone operator and chart

78

BREAKING

Dynamo Snapshot moves to its own operator and chart (Snapshot v0.1.0), installable separately or pulled into the platform chart via the new `global.snapshot.install=true` Helm value; `PodSnapshot` CRDs replace the removed `DynamoCheckpoint` CRD.

— Names Helm value and CRD swap but no migration walkthrough. v1.5.0

03 Triton backend for non-LLM models NEW

78

A new `dynamo.triton` backend, launched via `python -m dynamo.triton`, lets non-LLM models (vision, embedding, tensor-in/tensor-out) run inside a Dynamo pipeline alongside LLM workers.

Run a Triton-hosted vision or embedding model inside a Dynamo pipeline without switching to an LLM runtime.

```
$ python -m dynamo.triton
```

— Includes exact launch command and supported model types. v1.5.0

04 **Dynamo Sidecar for stock inference engines** NEW 73

Introduces Dynamo Sidecar (experimental): runs beside a stock inference engine over its native gRPC API in a separate process, with wheel-installed launchers for SGLang, TensorRT-LLM, and vLLM in one CPU-only multi-arch image.

— Explains architecture and packaging but no install command. v1.5.0

05 **CRD storage version promoted to v1beta1** BREAKING 70

Promotes `nvidia.com/v1beta1` to the CRD storage version for Kubernetes — the version the API server persists — and published examples drop their v1alpha1 copies, which must be migrated.

— Explains what storage version means and migration need. v1.5.0

06 **Single image decode at Frontend for multimodal pipelines**

IMPROVED 68

Enables image decoding once at Dynamo Frontend for encode-prefill-decode deployments on SGLang and vLLM (via libjpeg-turbo and FFmpeg), transferring pixels to encode workers over a single NIXL transfer.

— Explains mechanism and libraries but no config toggle. v1.5.0

07 **KV indexing extended to Mooncake and vLLM storage tiers**

IMPROVED 65

Extends KV indexing to Mooncake shared-cache events for SGLang HiCache (no HTTP queries on the hot path) and routes vLLM storage KV events to a disk tier with locality gating.

— Describes mechanism and scope but no config key given. v1.5.0

08 **Deployment-level thinking/reasoning controls** NEW 65

Adds an `enable_thinking` gate for Gemma 4 and deployment-level defaults for thinking/reasoning controls, resolving with a single precedence across all chat-template keys.

— Names flag and precedence model but no example. v1.5.0

09 **Typed error responses for over-context and overload requests**

IMPROVED 65

Rejects over-context requests with a typed 4xx before the stream opens; validation failures return HTTP 400 and backend overloads return HTTP 503.

— Names exact status codes but no endpoint path. v1.5.0

10 **TLS and mutual TLS on request/response and NATS transports** NEW

63

Adds opt-in TLS and mutual TLS on the TCP request/response planes and the NATS transport, with certificate paths injected once through `InfrastructureConfiguration`.

— Names config surface but gives no enablement steps. v1.5.0

11 **Operator runtime overrides, feature gates, and canary health checks**

IMPROVED

60

The Dynamo Operator adds `runtimeVersionOverride`, runtime feature gates, explicit multinode component roles, and default worker canary health checks.

— Names four new operator flags/behaviors without further mechanism. v1.5.0

12 **Experimental SGLang RL rollout contract** NEW

60

Adds an experimental SGLang RL rollout contract with detailed finish reasons, prompt token IDs and logprobs in `nvext`, native pause controls, and exact stop-token output.

— Names nvext fields but no usage example. v1.5.0

THINNER COVERAGE BELOW

13 **Pluggable worker selection policies in router** NEW

55

Introduces pluggable worker selection: native scorer and picker policies run through the same host-selection path as the built-in selector, with composable worker filters applying after eligibility and before scoring.

— Explains selection mechanism but no config name or flag. v1.5.0

14 **llm-d batch gateway example for offline OpenAI Batch requests**

NEW

55

Adds an llm-d batch gateway example for offline OpenAI Batch requests through the inference gateway, gated on Dynamo readiness.

— Names example and gating condition, no command shown. v1.5.0

15 **Pluggable tokenizer backends in Dynamo Frontend** NEW

50

Adds pluggable tokenizer backends to Dynamo Frontend, including a Baseten-contributed backend with a configurable Hugging Face fallback.

— Names contributed backend but no selection config. v1.5.0

16 **Streaming guided tool calls and Qwen3 tool-call parser** IMPROVED

50

Streams guided tool calls incrementally by default; adds Qwen3 support to the v2 unified tool-call parser.

— States behavior change without config specifics. v1.5.0

17 **Rust EPP replaces Go Endpoint Picker** BREAKING 50

Replaces the Go Endpoint Picker with the Rust EPP as the default, shipped inside the Dynamo Frontend image; the Go Endpoint Picker is removed.

— States removal but no migration steps. v1.5.0

18 **New deployment recipes for large models** NEW 50

Adds new recipes for Qwen3.8-2.4T-A95B FP8, Qwen3.5-122B FP8 on H200, Nemotron-3.5-Lightning, Nemotron-3-Ultra, and GLM-5.3-Flash on vLLM.

— Names models and configs, but no recipe details. v1.5.0

19 **Per-chunk token usage on /v1/messages streaming** IMPROVED 50

Emits per-chunk token usage on `/v1/messages` streaming responses.

— Names endpoint, minimal further detail. v1.5.0

20 **Observability additions: trace sink, readiness metric, ServiceMonitor labels** NEW 48

Writes request-trace records to a native S3 sink, adds a model readiness metric that gates batch dispatch, and allows custom ServiceMonitor labels in the Dynamo Helm chart.

— Groups three thin monitoring additions without config keys. v1.5.0

21 **Experimental GPU Memory Service V1** NEW 45

Adds an experimental GPU Memory Service V1 with a checkpoint lifecycle and engine integration for SGLang and vLLM.

— Named engines but no interface or API details. v1.5.0

22 **KVBM deprecated** DEPRECATED 45

KVBM is deprecated with removal targeted for v1.6.0; deployments relying on KVBM should migrate before then.

— States deprecation timeline but no migration path given. v1.5.0

23 **Conditional disaggregation bypass and session affinity in router**

NEW

43

Adds a conditional disaggregation bypass that sends short requests straight to decode, and adds session affinity in hard and soft forms to the Dynamo Router.

— Named additions without configuration or usage details. v1.5.0

24 **Video-aware KV hashing and audio/video inputs for vLLM workers**

NEW

43

Adds video-aware KV hashing for Qwen video inputs; vLLM workers now accept audio and video inputs.

— Named model support but no usage detail. v1.5.0

25 **Engine version updates** IMPROVED

40

Updates engines to SGLang v0.5.18, TensorRT-LLM v1.3.0rc25, and vLLM v0.28.0.

— Version bump only, no behavior change described. v1.5.0

26 **Realtime audio streaming and dynamic LoRA for aggregated workers**

NEW

35

Adds realtime audio streaming and dynamic LoRA for aggregated workers via vLLM-Omni.

— Brief description without mechanism or config. v1.5.0

└──▷ BREAKING ON UPGRADE

! `nvidia.com/v1beta1` is now the CRD storage version — the version the API server persists; v1alpha1 copies are removed from published examples and must be migrated.

! The Go Endpoint Picker is removed; the Rust EPP is now the only and default endpoint picker.

! The `DynamoCheckpoint` CRD is removed; `PodSnapshot` CRDs replace it in the standalone Dynamo Snapshot operator.

! AIConfigurator wheels and the `dynamo.replay` entrypoint are replaced by the `aisimulate` package at v0.12.0; existing scripts invoking `dynamo.replay` will break.

! KVBM is deprecated (removal targeted for v1.6.0) — deployments relying on KVBM should migrate before v1.6.0.

v1.5.0 NOTES ↗ LATEST STABLE

Dynamo v1.5.0 adds pluggable router policies, tiered KV indexing, TLS on all transports, a Triton backend, and a standalone Sidecar for stock inference engines.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.5.0 https://github.com/ai-dynamo/dynamo.git
# already have the repo? check out this version:
$ git checkout v1.5.0
```

└──▷ **TRY IT**

Run a Triton-hosted vision or embedding model inside a Dynamo pipeline without switching to an LLM runtime.

```
$ python -m dynamo.triton
```

Adds `runtimeVersionOverride`, runtime feature gates, explicit multinode component roles, and default worker canary health checks to the Dynamo Operator.

Adds `global.snapshot.install=true` Helm value to pull Dynamo Snapshot into the platform chart for evaluation, with `PodSnapshot` CRDs replacing the removed `DynamoCheckpoint` CRD.

Supports `python -m dynamo.triton` launcher to bring non-LLM models (vision, embedding, tensor-in/tensor-out) into a Dynamo pipeline alongside LLM workers via the new `dynamo.triton` backend.

Adds opt-in TLS and mutual TLS on the TCP request/response planes and the NATS transport, with certificate paths injected once through `InfrastructureConfiguration`.

Introduces pluggable worker selection: native scorer and picker policies run through the same host-selection path as the built-in selector, with composable worker filters applying after eligibility and before scoring.

+24 more

└──▷ **BREAKING ON UPGRADE**

- ! `nvidia.com/v1beta1` is now the CRD storage version — the version the API server persists; v1alpha1 copies are removed from published examples and must be migrated.
- ! The Go Endpoint Picker is removed; the Rust EPP is now the only and default endpoint picker.
- ! The `DynamoCheckpoint` CRD is removed; `PodSnapshot` CRDs replace it in the standalone Dynamo Snapshot operator.
- ! AIConfigurator wheels and the `dynamo.replay` entrypoint are replaced by the `aisimulate` package at v0.12.0; existing scripts invoking `dynamo.replay` will break.

> ! KVBM is deprecated (removal targeted for v1.6.0) — deployments relying on KVBM should migrate before v1.6.0.

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore

Feedback

changelog · Sep 22, 2026 | CHANGELOG ↗ | RANK

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

> Amazon Bedrock AgentCore's harness gained two new capabilities: custom OpenAI-compatible endpoint routing via a new `apiBase` field, and lifecycle hooks that can block or observe invocations and tool calls using Lambda, SNS, and EventBridge targets.

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

01 Lifecycle hooks for invocations and tool calls NEW

85

The AgentCore harness adds lifecycle hooks at `before_invocation`, `before_tool_call`, `after_tool_call`, and `after_invocation` boundaries, including for inline tool functions. AWS Lambda can be attached as a blocking target that returns a synchronous `allow` / `deny` decision to stop an invocation or skip a tool call, while Amazon SNS and Amazon EventBridge can be attached as non-blocking targets that receive notifications without altering agent-loop behavior. `InvokeHarness` now emits a `hookEvent` for each configured lifecycle hook that fires, carrying context for the corresponding lifecycle event.

— Names all hook boundaries, targets, and the hookEvent field.

changelog-20260922-54274c37 | Harness: Lifecycle hooks (2026-09-21)

02 Custom OpenAI-compatible endpoint routing via apiBase NEW

78

OpenAI model configurations in the harness gain an optional `apiBase` field to route requests through a custom OpenAI-compatible gateway, proxy, self-hosted endpoint, or regional endpoint instead of hitting openai.com directly.

Route a harness's model traffic through a self-hosted OpenAI-compatible proxy instead of the public OpenAI API.

json

```
{
  "model": "gpt-4o",
  "apiBase": "https://my-openai-proxy.internal/v1"
}
```

Route all OpenAI API calls through an internal gateway or regional proxy instead of hitting openai.com directly.

```json
{
  "model": "gpt-4o",
  "apiBase": "https://my-openai-proxy.internal/v1"
}
```

— Exact field name plus a runnable JSON config example.

Harness: Custom OpenAI-compatible endpo… changelog-20260922-54274c37

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Amazon Bedrock AgentCore adds custom OpenAI-compatible gateway routing and lifecycle hooks with Lambda/SNS/EventBridge targets.

└──▷ USE IT

Route all OpenAI API calls through an internal gateway or regional proxy instead of hitting openai.com directly.

```json
{
  "model": "gpt-4o",
  "apiBase": "https://my-openai-proxy.internal/v1"
}
```

Adds lifecycle hooks at `before_invocation`, `before_tool_call`, `after_tool_call`, and `after_invocation` boundaries — including for inline tool functions — letting you attach AWS Lambda targets that return a synchronous `allow` / `deny` decision to stop an invocation or skip a tool call.

Adds Amazon SNS and Amazon EventBridge as non-blocking lifecycle hook targets that receive notifications without altering agent-loop behavior.

`InvokeHarness` now emits a `hookEvent` for each configured lifecycle hook that fires, carrying context for the corresponding lifecycle event.

Harness: Custom OpenAI-compatible endpoints (2026-09-21) BLOG ↗ STABLE

Amazon Bedrock AgentCore harnesses now accept `apiBase` to route model traffic through any OpenAI-compatible endpoint.

└──▷ USE IT

Route a harness's model traffic through a self-hosted OpenAI-compatible proxy instead of the public OpenAI API.

```json
{
  "model": "gpt-4o",
  "apiBase": "https://my-openai-proxy.internal/v1"
}
```

Adds optional `apiBase` field to OpenAI model configurations to route requests through a custom OpenAI-compatible gateway, proxy, self-hosted endpoint, or regional endpoint.

Harness: Lifecycle hooks (2026-09-21) BLOG ↗ STABLE

Amazon Bedrock AgentCore harness gains lifecycle hooks for invocations and tool calls, backed by Lambda, SNS, or EventBridge.

WAS THIS USEFUL? 👍 👎

compressed-tensors

Feedback ⓘ 0.19.0 NOTES ↗ RANK

compressed-tensors extends safetensors to store and load quantized, sparse, and otherwise compressed model weights.

> compressed-tensors 0.19.0 focuses on scaling and reliability of the conversion pipeline, adding multi-GPU scheduling, distributed decompression, a MoE pruning converter, and several quantization-lifecycle breaking changes that push observer/qstrategy logic out of this repo.

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

01 Dynamic multi-GPU scheduler for conversion pipeline NEW

70

Adds a dynamic multi-GPU scheduler (`exec_jobs_dynamic` , `_pick_device` , `_free_bytes`) to the `entrypoints/convert` pipeline that schedules conversion work across accelerators based on measured free memory.

— Named functions and mechanism but no runnable example 0.19.0

02 TensorProfiler for peak memory measurement IMPROVED

70

Adds `TensorProfiler` (a `TorchDispatchMode`) to measure true peak memory during the converter validate pass, replacing the previous `x3` multiplier heuristic with a `2.5x` fallback estimate.

— Clear before/after mechanism with named class 0.19.0

03 MagnitudeExpertPruner converter for MoE pruning NEW

65

Adds a `MagnitudeExpertPruner` converter (`compressed_tensors.entrypoints.convert.converters.MagnitudeExpertPruner`) that structurally prunes MoE experts by magnitude, backed by new MoE pruning utilities under `compressed_tensors/utils/moe.py` .

— Names class path and utils file but no usage example 0.19.0

04 Memoized calculate_range for quantization IMPROVED

65

Memoizes `calculate_range` per (type, num_bits, device), eliminating a synchronous host-to-device copy per call — previously invoked ~245k times per 0.5B model under GPTQ W4A16.

— Concrete numbers and mechanism, no direct API for readers 0.19.0

05 Meta-device support for model-free PTQ NEW

60

Adds fast NVFP4 compression on meta-device tensors, skipping quantize+pack and producing correctly-shaped meta placeholders for model-free PTQ runs, plus qparam calculation on meta device enabling resumable model-free PTQ runs where already-processed shards are accounted as meta tensors without recomputing.

— Mechanism explained but no API entry point named `0.19.0`

06 **QuantizedKVCache structural read delegation** NEW 60

`QuantizedKVCache` now delegates structural reads (e.g. `past_key_values.layers[idx].keys`) to the wrapped `Cache`, enabling calibration of `transformers>=5` models that read the KV cache structurally.

— Names class, access pattern and transformers version target `0.19.0`

THINNER COVERAGE BELOW

07 **Quantization lifecycle simplification** BREAKING 55

GPTQ Group/Dynamic Activation Ordering code paths have been removed from the quantization lifecycle; automatic `qstrategy` inference has been removed so `qargs` must now always be passed explicitly; and observer resolution/defaulting has moved to llm-compressor, leaving `QuantizationArgs` in this repo simplified and no longer handling observer defaults.

— Migration note given but no code example of fix `0.19.0`

08 **Multi-converter chaining in converter interface** NEW 45

The converter interface now accepts a list of converters and chains validation and config updates through each, enabling pipelines such as prune-then-quantize.

— Behaviour described but no concrete API call shown `0.19.0`

09 **Distributed decompression in model_compressor** NEW 45

Adds distributed decompression support to `model_compressor`, running decompression hooks as a single thread under distributed execution.

— Names module and mechanism briefly, no example `0.19.0`

10 **transformers>=5.16 now required** BREAKING 40

`transformers>=5.16` is now required by compressed-tensors.

— Clear version bump but no rationale given `0.19.0`

11 **print_offload_folder utility** NEW 30

Adds `print_offload_folder`, a utility for inspecting offloaded model folders.

— Bare name, no usage detail given `0.19.0`

└──▷ **BREAKING ON UPGRADE**

! GPTQ Group/Dynamic Activation Ordering code paths have been removed from the quantization lifecycle.

! Automatic `qstrategy` inference removed; `qargs` must now always be passed explicitly.

! Observer resolution and defaulting moved to llm-compressor; `QuantizationArgs` in this repo is simplified and no longer handles observer defaults.

! `transformers>=5.16` is now required.

`0.19.0` NOTES ↗ **LATEST** STABLE

compressed-tensors 0.19.0 adds multi-GPU checkpoint conversion, MagnitudeExpertPruner, distributed decompression, and multi-converter chaining.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch 0.19.0 https://github.com/vllm-
project/compressed-tensors.git
# already have the repo? check out this version:
$ git checkout 0.19.0
```

Adds `MagnitudeExpertPruner` converter (`compressed_tensors.entrypoints.convert.converters.MagnitudeExpertPruner`) that structurally prunes MoE experts by magnitude, with new MoE pruning utilities under `compressed_tensors/utils/moe.py`.

Adds multi-converter chaining to the converter interface, letting pipelines accept a list of converters and chain validation and config updates through each (e.g. prune-then-quantize).

Adds a dynamic multi-GPU scheduler (`exec_jobs_dynamic`, `_pick_device`, `_free_bytes`) to the `entrypoints/convert` pipeline, scheduling conversion work across accelerators based on measured free memory.

Adds `TensorProfiler` (`TorchDispatchMode`) to measure true peak memory during the converter validate pass, replacing the previous `x3` multiplier heuristic with a `2.5x` fallback estimate.

Adds `print_offload_folder` utility for inspecting offloaded model folders.

+5 more

└──▷ **BREAKING ON UPGRADE**

- ! GPTQ Group/Dynamic Activation Ordering code paths have been removed from the quantization lifecycle.
- ! Automatic `qstrategy` inference removed; `qargs` must now always be passed explicitly.
- ! Observer resolution and defaulting moved to llm-compressor; `QuantizationArgs` in this repo is simplified and no longer handles observer defaults.
- ! `transformers>=5.16` is now required.

WAS THIS USEFUL? 👍 👎

Google Gemini Enterprise Agent Platform

Feedback

changelog · Sep 22, 2026 | CHANGELOG ↗ | RANK

Gemini Enterprise Agent Platform provides Google's managed model training, inference, evaluation, and agent deployment services, formerly offered under Vertex AI.

> Gemini Enterprise Agent Platform shipped reinforcement learning fine-tuning job management in the console, GA availability for Grok 4.6, Anthropic's Cyber Verification Program for Claude models, and a breaking migration to Agent Platform SDK for Python 2.0.1, alongside video streaming, Provisioned Throughput, Workbench scheduling, and CodeMender CLI updates.

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

01 **Reinforcement learning fine-tuning in console** NEW 95

Adds reinforcement learning fine-tuning job management in the Google Cloud console (`Models > Tuning` page), supporting Python code or model-based reward functions, pre-launch reward logic testing, real-time training metrics, and checkpoint testing in Agent Studio (Preview).

Create a reinforcement learning fine-tuning job for a Gemini model with a custom reward function via the Google Cloud console.

📍 1. In the Google Cloud console, navigate to Models › Tuning. 2. Click Create tuning job and select Reinforcement learning fine-tuning. 3. Choose your base Gemini model and define a Python code or model-based reward function. 4. Use Test reward logic to validate the function against sample prompts before submitting. 5. Submit the job and track training and evaluation metrics in real time. 6. When a checkpoint is ready, click Test in Agent Studio to evaluate the tuned model.

— Exact console path and step-by-step example provided changelog-20260922-ca003e64

02 **Scheduled notebook run configuration in Workbench** IMPROVED 85

Agent Platform Workbench scheduled notebook runs now require `scheduler-jupyter-plugin` version `0.1.8` or later to access new configuration options, support Shielded VM options including Secure Boot for verified boot integrity compliance, and accept a custom VM image or container as the runtime environment so scheduled runs can use the same dependencies as the interactive environment.

When scheduling a notebook run that must meet verified-boot compliance requirements, enable Shielded VM and Secure Boot through the Workbench scheduler UI.

📍 In Agent Platform Workbench, open a notebook and select `Schedule` › `Runtime environment`. Under `Shielded VM options`, enable `Secure Boot`, then confirm and save the schedule.

When a scheduled notebook must run with the same dependencies as your interactive session, specify a custom container image as the runtime environment.

📍 In Agent Platform Workbench, open a notebook and select `Schedule` › `Runtime environment`. Choose `Custom container`, enter your container image URI, then save the schedule.

— Exact plugin version and UI steps given for each option `September 09, 2026 (2026-09-09)`

---

03 **CodeMender CLI configuration and scan safety limits** IMPROVED

85

CodeMender CLI adds `scan_config.exclude_dirs` as the standardized configuration key for directory exclusion rules across config files and CLI scanning flags (v0.7.0), sets `gemini-3.8-flash` as the new default model for CLI sessions with a one-time in-CLI notice when the new default activates (v0.8.0), and introduces safe output limits for file reading (2 MiB) and codebase grep search (512 KiB) with centered match context windows to prevent payload overflow during large repository scans (v0.8.0).

Standardize directory exclusions across all CodeMender scans so that vendor and build directories are never traversed.

yaml

```yaml
scan_config:
  exclude_dirs:
    - vendor
    - node_modules
    - build
    - .git
```

— Groups three CLI additions with a config example, no flag names given

`changelog-20260922-ca003e64`

---

04 **Agent Platform SDK for Python 2.0.1** BREAKING

80

Introduces `google-cloud-agentplatform` version 2.0.1, migrating generative AI modules to the Google Gen AI SDK and decoupling the agent surface from `google-cloud-aiplatform` into a dedicated package with restructured namespaces; existing code importing from `google-cloud-aiplatform` requires migration per the official migration guide.

— Names package and breaking change but no migration steps shown

changelog-20260922-ca003e64

05 **Stateful storage and streaming in Interactions API** NEW

75

The Interactions API adds `store: true` and `stream: true` parameters, enabling stateful server-side video storage and server-sent event (SSE) streaming for video generation with Gemini Omni Flash (Preview).

— Named parameters given but no endpoint path or sample request

changelog-20260922-ca003e64

06 **Anthropic Cyber Verification Program support** NEW 65

Enables Anthropic's Cyber Verification Program (CVP) on Gemini Enterprise Agent Platform (Preview), allowing verified organizations to use Claude Opus 4.7, Claude Opus 4.8, Claude Sonnet 5, and Claude Opus 5 for defensive cybersecurity tasks with dual-use restrictions lifted.

— Names models and restriction change but no enrollment steps changelog-20260922-ca003e64

THINNER COVERAGE BELOW

07 **Client-side log forwarding for JupyterLab** NEW 55

JupyterLab now forwards client-side logs — console errors, uncaught exceptions, unhandled promise rejections, and failed network requests — to the instance backend, surfacing them in Cloud Logging.

— Describes behavior but no config or query example September 21, 2026 (2026-09-21)

08 **Provisioned Throughput model and order scheduling updates**

IMPROVED 55

Adds Gemini Omni to Provisioned Throughput supported models, and enables Provisioned Throughput to schedule multiple new orders and change requests across multiple orders simultaneously within the same project via the self-service console.

— Names console feature but no exact UI path or API changelog-20260922-ca003e64

09 **Grok 4.6 general availability** NEW 45

Adds xAI's Grok 4.6 as a generally available model on the global endpoint and the US multi-region endpoint.

— No usage example or model identifier given changelog-20260922-ca003e64

└──▷ **BREAKING ON UPGRADE**

! Agent Platform SDK for Python 2.0.1 (`google-cloud-agentplatform`) decouples the agent surface from `google-cloud-aiplatform` into a dedicated package with restructured

namespaces — existing code importing from `google-cloud-aiplatform` requires migration per the Agent Platform SDK for Python version 2.0.1 migration guide.

changelog · Sep 22, 2026 | CHANGELOG ↗ | LATEST | STABLE

Gemini Enterprise Agent Platform adds Grok 4.6 GA, Python SDK 2.0.1, SSE video streaming, RL fine-tuning UI, and CVP for Claude.

└──▷ USE IT

Standardize directory exclusions across all CodeMender scans so that vendor and build directories are never traversed.

.codemender/config.yaml

```
scan_config:
  exclude_dirs:
    - vendor
    - node_modules
    - build
    - .git
```

Create a reinforcement learning fine-tuning job for a Gemini model with a custom reward function via the Google Cloud console.

📍 1. In the Google Cloud console, navigate to Models › Tuning. 2. Click Create tuning job and select Reinforcement learning fine-tuning. 3. Choose your base Gemini model and define a Python code or model-based reward function. 4. Use Test reward logic to validate the function against sample prompts before submitting. 5. Submit the job and track training and evaluation metrics in real time. 6. When a checkpoint is ready, click Test in Agent Studio to evaluate the tuned model.

Adds `store: true` and `stream: true` parameters to the Interactions API, enabling stateful server-side video storage and server-sent event (SSE) streaming for video generation with Gemini Omni Flash (Preview).

Adds `scan_config.exclude_dirs` as the standardized configuration key for directory exclusion rules across CodeMender configuration files and CLI scanning flags (v0.7.0).

Introduces Agent Platform SDK for Python version 2.0.1 (`google-cloud-agentplatform`), migrating generative AI modules to the Google Gen AI SDK, decoupling the agent surface from `google-cloud-aiplatform` into a dedicated package, and restructuring namespaces.

Adds xAI's Grok 4.6 as a generally available model on the global endpoint and the US multi-region endpoint.

Adds reinforcement learning fine-tuning job management in the Google Cloud console (`Models > Tuning` page), supporting Python code or model-based reward functions,

pre-launch reward logic testing, real-time training metrics, and checkpoint testing in Agent Studio (Preview).

+5 more

└──▷ **BREAKING ON UPGRADE**

! Agent Platform SDK for Python 2.0.1 (`google-cloud-agentplatform`) decouples the agent surface from `google-cloud-aiplatform` into a dedicated package with restructured namespaces — existing code importing from `google-cloud-aiplatform` requires migration per the Agent Platform SDK for Python version 2.0.1 migration guide.

September 21, 2026 (2026-09-21) BLOG ↗ STABLE

JupyterLab client-side logs now forwarded to Cloud Logging for centralized visibility

September 09, 2026 (2026-09-09) BLOG ↗ STABLE

Agent Platform Workbench v2 scheduled notebook runs gain execution identity, Shielded VM, and custom runtime options

└──▷ **HOW TO FIND IT**

When scheduling a notebook run that must meet verified-boot compliance requirements, enable Shielded VM and Secure Boot through the Workbench scheduler UI.

📍 In Agent Platform Workbench, open a notebook and select Schedule › Runtime environment . Under Shielded VM options , enable Secure Boot , then confirm and save the schedule.

When a scheduled notebook must run with the same dependencies as your interactive session, specify a custom container image as the runtime environment.

📍 In Agent Platform Workbench, open a notebook and select Schedule › Runtime environment . Choose Custom container , enter your container image URI, then save the schedule.

Requires `scheduler-jupyter-plugin` at version `0.1.8` or later to access the new scheduled-run configuration options.

Scheduled notebook runs now support Shielded VM options, including Secure Boot, to meet workload requirements that mandate verified boot integrity.

Scheduled notebook runs now accept a custom VM image or container as the runtime environment, allowing scheduled runs to use the same dependencies as your interactive environment.

WAS THIS USEFUL? 👍 👎

Semantica AGI Semantica

Feedback ⓘ v0.7.0 NOTES ↗

RANK

Semantica provides graph-native context infrastructure for building accountable AI systems.

> Semantica's v0.7.0 release and accompanying docs updates add a decision-provenance graph with causal ancestry and policy gating to ContextGraph, Hierarchical Community GraphRAG search, six new data connectors, schema-guided extraction, and MCP retrieval tools, alongside breaking changes including a Python 3.10+ requirement and a new PolicyEngine error-handling contract.

└──▷ **WHAT SHIPPED · 20 FEATURES** most completely described first ? what's the number?

01 **Decision provenance graph in ContextGraph** NEW 93

`ContextGraph` (from `semantica.context`) provides structured, queryable agent memory via `add_node`, `add_edge`, `get_neighbors`, and point-in-time `state_at` snapshots, with an `advanced_analytics` flag. record_decision() creates permanent decision records with `category`, `scenario`, `reasoning`, `outcome`, `confidence`, and `metadata` fields, exportable as W3C PROV-O; add_causal_relationship() links decisions with typed edges like `CAUSED` and `INFLUENCED` to build causal ancestry graphs; trace_decision_chain() retrieves that ancestry by ID; find_similar_decisions() (with a `max_results` parameter) surfaces precedent decisions by semantic query; analyze_decision_impact() maps downstream influence; check_decision_rules() gates decisions against policy rules by category and field values; and get_decision_insights() returns aggregate intelligence across all recorded decisions.

Record a chain of AI decisions with causal links and verify them against policy — useful for regulated workflows where every choice must be auditable.

```python
from semantica.context import ContextGraph

graph = ContextGraph(advanced_analytics=True)

app_id = graph.record_decision(
    category="credit_application",
    scenario="Personal loan, $85k income, 31% DTI, 3yr
employment",
    reasoning="Income meets threshold; employment stable; no
adverse credit events",
    outcome="proceed_to_underwriting",
    confidence=0.91,
    metadata={"applicant_id": "A-7291"},
)

uw_id = graph.record_decision(
    category="loan_underwriting",
    scenario="Underwriting review for A-7291",
    reasoning="DTI within policy; clean 36-month credit
history",
    outcome="approved",
    confidence=0.95,
)

graph.add_causal_relationship(app_id, uw_id,
relationship_type="CAUSED")

compliant = graph.check_decision_rules({"category":
"loan_underwriting", "confidence": 0.9})
chain = graph.trace_decision_chain(uw_id)
```

Record a high-stakes vendor decision with confidence score, then pull its full causal ancestry and check it against policy rules — useful for compliance audit trails in regulated environments.

```python
from semantica.context import ContextGraph

graph = ContextGraph(advanced_analytics=True)
decision_id = graph.record_decision(
    category="vendor_selection",
    scenario="Choose cloud provider for HIPAA workload",
    reasoning="AWS offers BAA, mature HIPAA tooling, and
existing team expertise",
    outcome="selected_aws",
    confidence=0.93,
)
chain = graph.trace_decision_chain(decision_id)
compliant = graph.check_decision_rules({"category":
"vendor_selection"})
print(chain, compliant)
```

Search past decisions by topic to surface precedents before making a new consequential choice — useful when an agent or analyst needs to know what was decided before on a similar scenario.

```python
from semantica.context import ContextGraph

graph = ContextGraph()
similar = graph.find_similar_decisions("cloud vendor",
max_results=5)
for d in similar:
    print(d)
```

— Full method list and fields, with runnable code examples. product docs

02 **Unified vector+graph retrieval via AgentContext** NEW

80

`AgentContext` (from `semantica.context`) combines a `VectorStore` backend (e.g. `faiss`) with a `ContextGraph` into one retrieval interface, exposing store() to write content and retrieve() to query across embedding similarity and graph traversal.

Give an agent a combined vector + graph memory so it can retrieve by embedding similarity AND graph traversal, with full provenance on every fact.

```python
from semantica.context import ContextGraph, AgentContext
from semantica.vector_store import VectorStore

graph = ContextGraph(advanced_analytics=True)
graph.add_node("acme_corp", "Organization", name="Acme Corp",
industry="SaaS")
graph.add_node("alice_chen", "Person", name="Alice Chen",
role="CTO")
graph.add_edge("alice_chen", "acme_corp", edge_type="works_for",
since="2019-03-01")

vs = VectorStore(backend="faiss")
ctx = AgentContext(vector_store=vs, knowledge_graph=graph)
ctx.store("Alice approved the Acme renewal in Q1 2024",
conversation_id="conv_001")
result = ctx.retrieve("who approved the Acme contract?")
```

— Named classes and methods with a runnable example. product docs

03 **Progress-bar control via environment variables** NEW

80

`SEMANTICA_DISABLE_PROGRESS` and `SEMANTICA_FORCE_PROGRESS` environment variables control progress-bar output in CI/pipe contexts, with `SEMANTICA_DISABLE_PROGRESS` taking precedence; e.g. `SEMANTICA_DISABLE_PROGRESS=1 python run_pipeline.py` silences bars when stdout is redirected.

Silence progress bars in CI pipelines where stdout is redirected to a log file.

```
$ SEMANTICA_DISABLE_PROGRESS=1 python run_pipeline.py
```

Silence progress bars in a CI pipeline where stdout is redirected, preventing garbled log output.

```
$ SEMANTICA_DISABLE_PROGRESS=1 python run_pipeline.py
```

— Exact env vars with a runnable shell command. product docs

04 **Packaging and Python version requirement changes** BREAKING

75

The `dev` extra is replaced by a PEP 735 dependency group — install with `pip install -e . --group dev` instead of `pip install 'semantica[dev]'`; `semantica[crewai]` no longer installs `crewai` automatically; and Python 3.8/3.9 are no longer supported, requiring Python 3.10+.

— Exact install commands and version cutoffs given. v0.7.0

05 **Provenance tracking and multi-format export** NEW 70

`ProvenanceManager` (from `semantica.provenance`) and RDFExporter (from `semantica.export`) track W3C PROV-O lineage on every fact and export it to RDF, OWL, Parquet, Cypher, and JSON-LD; data can also be exported to JSON and CSV, with point-in-time graph snapshots enabling time-travel queries.

— Names modules and formats but no call example. product docs

06 **Multi-source ingestion and extraction pipeline** NEW 70

Supports ingestion from files, web, Kafka, Kinesis, Git, email, Google Drive, Elasticsearch, and MCP, with NER/relation/event extraction, entity-aware chunking, semantic deduplication, and conflict detection.

— Names sources and extraction stages, no code path given. product docs

07 **Polyglot graph storage backends** NEW 65

Graph storage is swappable without code changes across RDF triple stores (Oxigraph, Blazegraph, Apache Jena, Eclipse RDF4J) and Labeled Property Graphs (Neo4j, FalkorDB, Apache AGE, AWS Neptune).

— Names all eight backends but no swap mechanism shown. product docs

08 **New data connectors for warehouses and lakehouses** NEW 65

Adds connectors for BigQuery, Amazon Redshift, Power BI, Apache Airflow, and line-delimited JSON ingestion, plus native connectors for Databricks (Unity Catalog + Delta Lake), Snowflake, and SAP OData, turning lakehouse and warehouse data into provenance-tracked graph nodes.

— Names all connectors across both releases, no setup steps. v0.7.0

09 **Hierarchical Community GraphRAG search** NEW 65

Adds multi-level Louvain/Leiden community detection, LLM-summarized community reports, and global Map-Reduce + DRIFT hybrid search for hierarchical community-based GraphRAG.

— Explains mechanism but no API or config surface named. v0.7.0

10 **Deterministic reasoning via SPARQL and Datalog** IMPROVED

65

Activates `SPARQLReasoner.execute_query` to run real SPARQL queries (previously a no-op), part of broader deterministic reasoning support via forward chaining, a Rete network, Datalog, and SPARQL, with SHACL constraints and OWL/SKOS ontology management.

– Names the fixed method and reasoning mechanisms, no example. v0.7.0

---

11 **MCP semantic retrieval tools** NEW

65

Adds MCP semantic retrieval tools `store_document`, `retrieve_context`, `update_document`, and `remove_document` for document lifecycle management over the MCP protocol.

– Names all four MCP tools, no call example provided. v0.7.0

---

12 **PolicyEngine raises on unevaluable rules** BREAKING

65

`PolicyEngine.check_compliance` now raises `ProcessingError` instead of returning `False` for unevaluable rules; callers that checked for a `False` return must now catch `ProcessingError`.

– Names method, old and new behavior, migration action. v0.7.0

---

13 **Schema-guided entity extraction** NEW

60

`SchemaValidator` and `bootstrap_schema` let callers constrain what the extractor produces to a declared schema for schema-guided entity extraction.

– Names classes/functions but no usage example. v0.7.0

---

14 **Native vector deletion in FAISS and Milvus** NEW

60

Adds `delete_vectors` to `MilvusStore` for native vector deletion, and adds equivalent native vector deletion support to the FAISS vector store.

– Names method and both stores, no code sample given. v0.7.0

THINNER COVERAGE BELOW

15 **Agent framework integrations and MCP/REST/CLI surfaces** NEW

55

Adds drop-in integrations for Agno, CrewAI, and LangChain agent frameworks, plus a full MCP server, REST API, and CLI.

– Names frameworks and surfaces but no endpoints or commands. product docs

---

16 **Entity.confidence becomes optional** BREAKING

55

`Entity.confidence` is now `Optional[float]` (was `float = 1.0`); code that unconditionally uses this value as a float needs a None guard.

— Exact before/after type with migration note. v0.7.0

17 **Persistent SQLite extraction cache** NEW 50

Adds a persistent SQLite backend for `ExtractionCache`, letting extraction results survive process restarts.

— Names the class and backend, no config shown. v0.7.0

18 **GraphBuilder for bi-temporal knowledge graphs** NEW 45

`GraphBuilder` constructs bi-temporal knowledge graphs from ingested data with full graph analytics including centrality, community detection, and link prediction.

— Describes capability without configuration or call detail. product docs

19 **Vendor-neutral LLM integration module** NEW 35

`semantica.llms` provides vendor-neutral LLM integration across OpenAI, Anthropic, Gemini, and other major providers.

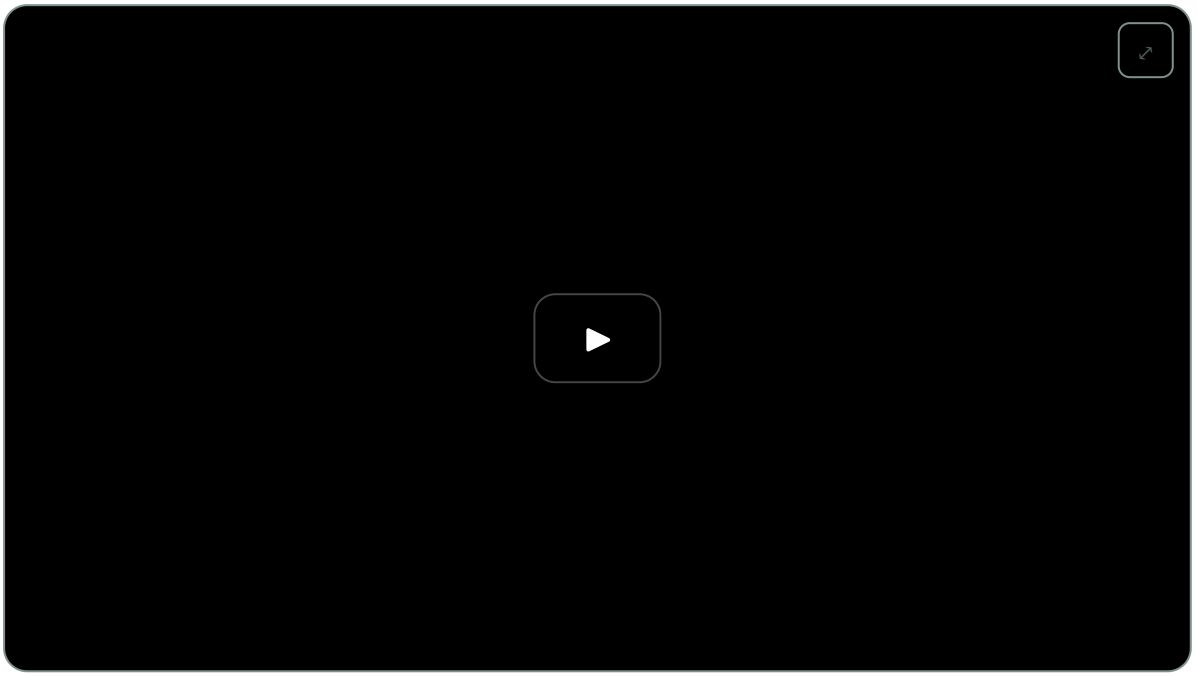
— Names module and providers but no usage detail. product docs

20 **Source-aware truth maintenance** NEW 35

Adds source-aware truth maintenance with corroboration-based trust tiers for graph facts.

— One-line description with no named interface. v0.7.0

└──▷ ALSO FROM THESE RELEASES



└──▷ BREAKING ON UPGRADE

! `Entity.confidence` is now `Optional[float]` (was `float = 1.0`); any code that unconditionally uses this value as a float will need a None guard.

! `PolicyEngine.check_compliance` now raises `ProcessingError` instead of returning `False` for unevaluable rules; callers that checked for a `False` return must now catch `ProcessingError`.

! The `dev` extra is replaced by a PEP 735 dependency group; install with `pip install -e . --group dev` instead of `pip install 'semantica[dev]'`.

! `semantica[crewai]` no longer installs `crewai` automatically.

! Python 3.8 and 3.9 are no longer supported; Python 3.10+ is required.

v0.7.0 NOTES ↗ LATEST STABLE

Semantica v0.7.0 adds GraphRAG community search, six data connectors, schema-guided extraction, MCP retrieval tools, and a persistent cache backend.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.7.0 https://github.com/semantica-agi/semantica.git
# already have the repo? check out this version:
$ git checkout v0.7.0
```

Adds MCP semantic retrieval tools `store_document`, `retrieve_context`, `update_document`, and `remove_document` for document lifecycle management over the MCP protocol.

Adds `SchemaValidator` and `bootstrap_schema` for schema-guided entity extraction, letting callers constrain what the extractor produces to a declared schema.

Adds a persistent SQLite backend for `ExtractionCache`, enabling extraction results to survive process restarts.

Adds six new data connectors: BigQuery, Amazon Redshift, Power BI, Apache Airflow, and line-delimited JSON ingestion.

Adds Hierarchical Community GraphRAG with multi-level Louvain/Leiden community detection, LLM-summarized community reports, and global Map-Reduce + DRIFT hybrid search.

+4 more

└──▷ BREAKING ON UPGRADE

! `Entity.confidence` is now `Optional[float]` (was `float = 1.0`); any code that unconditionally uses this value as a float will need a None guard.

! `PolicyEngine.check_compliance` now raises `ProcessingError` instead of returning `False` for unevaluable rules; callers that checked for a `False` return

must now catch `ProcessingError`.

! The `dev` extra is replaced by a PEP 735 dependency group; install with `pip install -e . --group dev` instead of `pip install 'semantica[dev]'`.

! `semantica[crewai]` no longer installs `crewai` automatically.

! Python 3.8 and 3.9 are no longer supported; Python 3.10+ is required.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

LM Studio

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗ RANK

LM Studio is a desktop application that runs local LLMs and provides model discovery, downloads, chat, and a local API server.

> LM Studio 0.4.25 adds support for the Splash engine on newer Apple Silicon hardware.

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

01 **Splash engine support on Apple Silicon** NEW 45

LM Studio adds support for the Splash engine on Apple Silicon M3 or newer running macOS 26.4 or newer.

— Names engine and hardware/OS requirements but no usage detail

changelog-20260922-46d9d82e

> changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE
>
> LM Studio 0.4.25 adds Splash engine support on Apple Silicon M3+ with macOS 26.4+.

WAS THIS USEFUL? 👍 👎

vLLM

Feedback ⓘ v0.30.0 NOTES ↗ CODE ↗ RANK

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

> vLLM v0.30.0 focuses on restart and cold-start speed (IPC weight cache, faster CUDA graph capture), adds Gumbel-max watermarking and a host-resident sparse-MLA KV offload tier, brings in eight new models, and moves scale-out serving and the gRPC entrypoint behind explicit opt-in flags.

└──▷ **WHAT SHIPPED · 27 FEATURES** most completely described first ? what's the number?

01 Fast Start IPC weight cache NEW

88

Adds `--load-format ipc_cache` to enable a persistent per-GPU weight-cache daemon that maps post-quantized, TP-sharded weights over CUDA IPC on engine restart, covering FP4 checkpoints and multi-node TP, eliminating disk reloads on restart.

Restart a vLLM engine instantly by mapping pre-cached weights over CUDA IPC instead of reloading from disk — cuts cold-start time dramatically in multi-restart workflows.

```
$ vllm serve meta-llama/Llama-3.1-8B-Instruct --load-format ipc_cache
```

— Named flag, mechanism, and scope (FP4, multi-node TP) all given. v0.30.0

02 Scale-out endpoints now opt-in BREAKING

81

`vllm serve` now requires the `--enable-scale-out` flag to activate scale-out endpoints; the previous `VLLM_ENABLE_SCALE_OUT_ENDPOINTS` environment variable no longer activates them.

Serve a model with scale-out disaggregated endpoints enabled for prefill/decode separation in large-scale deployments.

```
$ vllm serve deepseek-ai/DeepSeek-V4 --enable-scale-out
```

— Exact flag, replaced env var, and a runnable example given. v0.30.0

03 Multimodal media_io_kwargs additions NEW

79

Adds an `audio_backend` key in `--media-io-kwargs` for multimodal serving, with torchcodec audio decoding support and soundfile-first automatic decoding, plus a

`media_io_kwargs` field in multimodal hashes so cache hash kwargs are scoped by modality.

Serve a multimodal model selecting torchcodec as the audio backend for improved audio decoding performance.

```
$ vllm serve mistralai/Voxtral-Mini-3B-2507 --media-io-kwargs
'{"audio_backend": "torchcodec"}'
```

— Names the config key, sub-key, and cache field with an example. v0.30.0

---

04 **HiSparse host-resident sparse-MLA decode tier** NEW 78

Adds a host-resident sparse-MLA decode tier via `HiSparseConnector`, spilling KV pages to pinned host memory under GPU pressure with per-request GPU hot buffering and Prometheus counters.

— Names connector, mechanism, and monitoring surface clearly. v0.30.0

---

05 **Gumbel-max watermarked generation and detection** NEW 75

Adds Gumbel-max watermarked generation and detection with a keyed PRF, per-request opt-out, an example detection endpoint, and dual-key mode for compatibility with speculative decoding.

— Mechanism and modes described but no concrete endpoint/flag shown. v0.30.0

---

06 **CUDA graph capture and engine init speedups** IMPROVED 75

Reduces CUDA graph capture time from 12s to 2s and engine init from 28.9s to 8.2s on H200 by freezing GC during graph capture.

— Concrete before/after numbers and mechanism, no flag to invoke. v0.30.0

---

07 **gRPC server entrypoint replaced** DEPRECATED 70

Adds `--grpc` flag to `vllm serve` as the replacement entrypoint for the gRPC server, deprecating `python -m vllm.entrypoints.grpc_server`.

— Both old and new entrypoints named exactly. v0.30.0

08 **Quantization backend additions** IMPROVED 65

Adds AutoRound 2/3/5/6/7-bit quantization on CUDA, W4A16 DSA with the `nvfp4_fp8_ds_mla` KV cache format, per-quantization linear backend overrides, and NVFP4 support in the torch linear backend; makes FlashInfer CuTeDSL NVFP4 W4A16 the default over Marlin on SM100/103.

— Names formats and a default-behaviour change across several bit-widths. v0.30.0

09 **Sampling mask return for RL workloads** IMPROVED 62

Adds `--return-sampling-mask` flag to return GPU-compacted sampling masks, fixing an approximately 2x RL step-time regression.

— Names flag and the regression fixed, no usage detail. v0.30.0

10 **Engram tensor parallelism for Qwen3.8-Flash-Next** NEW 62

Adds `--engram-config` flag to enable UVA PLE offload and Engram tensor parallelism for Qwen3.8-Flash-Next.

— Named flag and target model, mechanism only briefly named. v0.30.0

11 **Targeted online quantization via quantization_config.targets** NEW 62

Adds `quantization_config.targets` to enable targeted online quantization and supports partial pre-quantized checkpoints from any quantization method.

— Named config field, no usage example given. v0.30.0

12 **QSA indexer memory limit control** NEW 62

Adds `VLLM_SPARSE_INDEXER_MAX_LOGITS_MB` environment variable to control separate prefill and decode QSA indexer kernel memory limits for Qwen3.8-Flash-Next.

— Named env var and scope, no configuration example. v0.30.0

13 **Default fp8_ds_mla KV cache resolution** IMPROVED 62

`--kv-cache-dtype auto` now resolves to `fp8_ds_mla` when FlashMLA is active for DeepSeek V4.

— Named flag and resolved value, clear condition given. v0.30.0

14 **DeepSeek-V4 CPU backend** NEW 60

Adds a DeepSeek-V4 CPU backend with AVX512/AMX sparse MLA, indexer, mHC, and compressor kernels.

— Names kernel set but no flag or invocation given. v0.30.0

15 **Speculative decoding improvements in Model Runner V2** IMPROVED

60

Adds MTP and EAGLE3/DFlash/DSpark speculative decoding support under pipeline parallelism, adaptive verification for every draft-model speculator through an online acceptance estimator, and dual-batch overlap in eager mode and with full CUDA graphs for microbatched steps.

— Names three mechanisms but no config surface to invoke them. v0.30.0

THINNER COVERAGE BELOW

16 **FlashInfer PCIe IPC all-reduce** NEW

55

Adds an opt-in FlashInfer PCIe IPC all-reduce for NVLink-less multi-GPU boxes.

— Named mechanism and hardware scope, opt-in flag not named. v0.30.0

17 **New model support: DeepSeek, GLM, K2-Horizon and more** NEW

53

Adds support for DeepSeek-V4.1-Flash, DeepSeek-V4-Flash-Vision-Exp (including ROCm and LoRA), GLM-5.3-Flash, K2-Horizon (with new reasoning and tool parsers), Cohere Compass, Bailing V3 VL with MTP, and Nanbeige4.2 via the Transformers backend.

— Enumerates all models but no usage mechanism per model. v0.30.0

18 **Encoder-cache sharing over NIXL and Mooncake** NEW

50

Adds encoder-cache sharing over NIXL and Mooncake for large-scale serving.

— Names two integrated systems but no config detail. v0.30.0

19 **Deprecated environment variables removed** BREAKING

50

The `VLLM_PREFIX_CACHE_RETENTION_INTERVAL` and `VLLM_MM_HASHER_ALGORITHM` environment variables, deprecated in 0.29, have been removed.

— Names both variables and prior deprecation version. v0.30.0

20 **YaRN alignment with Transformers changes context length**

48

BREAKING

Vendor YaRN aliases no longer re-scale `max_model_len`, which may change effective context lengths for models using those aliases.

— Explains the behaviour change but no migration steps. v0.30.0

21 **Encoder CUDA graphs for MiniCPM-V** NEW

47

Adds encoder CUDA graphs for MiniCPM-V 2.5, 2.6, and 4.0.

— Names exact model versions but no further mechanism. v0.30.0

22 **Voxtral Realtime support** NEW 45

Adds Voxtral Realtime support with `FULL_DECODE_ONLY` graphs.

— Names the graph mode but no configuration path. v0.30.0

23 **Elastic EP CUDA graph reuse** IMPROVED 43

Adds Elastic EP reuse of CUDA graphs across EP reconfiguration.

— Bare description of the behaviour change. v0.30.0

24 **KVCR secondary-tier KV cache adapter** NEW 35

Adds a KVCR secondary-tier KV cache adapter.

— Names the adapter only, no mechanism given. v0.30.0

25 **DeepSelect top-k for DSA sparse indexer** NEW 35

Adds DeepSelect top-k for the DSA sparse indexer.

— One-line addition with no mechanism or config named. v0.30.0

26 **GPTQ activation ordering removed** BREAKING 35

GPTQ activation ordering (`g_idx`) support has been removed.

— States the removal only, no migration guidance. v0.30.0

27 **Mamba 'all' cache mode deprecated** DEPRECATED 25

The `all` Mamba cache mode is deprecated.

— Bare deprecation notice with no replacement named. v0.30.0

└──▷ BREAKING ON UPGRADE

! Scale-out endpoints are now opt-in via `--enable-scale-out` on `vllm serve` ; the `VLLM_ENABLE_SCALE_OUT_ENDPOINTS` environment variable no longer activates them.

! GPTQ activation ordering (`g_idx`) support has been removed.

! The `VLLM_PREFIX_CACHE_RETENTION_INTERVAL` and `VLLM_MM_HASHER_ALGORITHM` environment variables (deprecated in 0.29) have been removed.

! YaRN alignment with Transformers means vendor YaRN aliases no longer re-scale `max_model_len` , which may change effective context lengths for models using those aliases.

! `python -m vllm.entrypoints.grpc_server` is deprecated in favor of `vllm serve --grpc` .

! The `all` Mamba cache mode is deprecated.

v0.30.0 NOTES ↗ CODE ↗ LATEST STABLE

vLLM v0.30.0 adds Fast Start IPC weight cache, Gumbel-max watermarking, HiSparse KV offload, 8 new models, and opt-in scale-out endpoints.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.30.0 https://github.com/vllm-project/vllm.git
# already have the repo? check out this version:
$ git checkout v0.30.0
```

└──▷ TRY IT

Restart a vLLM engine instantly by mapping pre-cached weights over CUDA IPC instead of reloading from disk — cuts cold-start time dramatically in multi-restart workflows.

```
$ vllm serve meta-llama/Llama-3.1-8B-Instruct --load-format ipc_cache
```

Serve a model with scale-out disaggregated endpoints enabled for prefill/decode separation in large-scale deployments.

```
$ vllm serve deepseek-ai/DeepSeek-V4 --enable-scale-out
```

Serve a multimodal model selecting torchcodec as the audio backend for improved audio decoding performance.

```
$ vllm serve mistralai/Voxtral-Mini-3B-2507 --media-io-kwargs '{"audio_backend": "torchcodec"}'
```

Adds `--load-format ipc_cache` to enable a persistent per-GPU weight-cache daemon that maps post-quantized, TP-sharded weights over CUDA IPC on engine restart, covering FP4 checkpoints and multi-node TP — eliminating disk reloads.

Adds `--return-sampling-mask` flag to return GPU-compacted sampling masks, fixing an approximately 2x RL step-time regression.

Adds `--engram-config` flag to enable UVA PLE offload and Engram tensor parallelism for Qwen3.8-Flash-Next.

Adds `--enable-scale-out` flag to `vllm serve` to opt in to scale-out endpoints, replacing the `VLLM_ENABLE_SCALE_OUT_ENDPOINTS` environment variable.

Adds `--grpc` flag to `vllm serve` as the replacement entrypoint for the gRPC server

(deprecating `python -m vllm.entrypoints.grpc_server`).

+26 more

└──▷ BREAKING ON UPGRADE

- ! Scale-out endpoints are now opt-in via `--enable-scale-out` on `vllm serve`; the `VLLM_ENABLE_SCALE_OUT_ENDPOINTS` environment variable no longer activates them.
- ! GPTQ activation ordering (`g_idx`) support has been removed.
- ! The `VLLM_PREFIX_CACHE_RETENTION_INTERVAL` and `VLLM_MM_HASHER_ALGORITHM` environment variables (deprecated in 0.29) have been removed.
- ! YaRN alignment with Transformers means vendor YaRN aliases no longer re-scale `max_model_len`, which may change effective context lengths for models using those aliases.
- ! `python -m vllm.entrypoints.grpc_server` is deprecated in favor of `vllm serve --grpc`.
- ! The `all` Mamba cache mode is deprecated.

WAS THIS USEFUL? 👍 👎

Open WebUI

Feedback ⓘ v0.11.4 NOTES ↗ RANK

Open WebUI is a self-hosted chat interface for local and remote models, with RAG over uploaded documents, tool calling, and multi-user access control.

> Open WebUI's headline addition is agentic terminal execution — agents get a real terminal and filesystem, with an integrated file browser, diffing, and per-terminal isolation — alongside a new slim Docker image for lightweight, external-model deployments and a batch of RAG, web search, skills, and admin UI improvements.

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

01 Agentic terminal execution with integrated file browser NEW

93

Open WebUI's Agentic Execution with Open Terminal gives agents a real terminal and filesystem to run scripts, analyze data, fix errors, and return produced files inside the chat interface, with a Terminals enterprise feature providing every user an isolated environment with resource limits and automatic lifecycle management. A sidebar file browser appears automatically when a terminal is connected, letting users browse, preview, upload, download, create, delete, rename, and edit files, controlled by Show Files on Terminal Select in Settings → Interface (per-user, on by default) and Terminal File Display (Sidebar or Inline) for where AI-generated files appear. Terminal Contexts give each conversation its own isolated workspace, appearing only after the first message is sent. The Chat Uploads connection setting can be set to Filesystem to route chat-input attachments straight to the terminal's working directory instead of extracting text for the model. Previews cover text/code (syntax highlighting, line numbers), PDFs, spreadsheets (CSV/TSV as tables), Word `.docx` (pagination, headers, footers, images), PowerPoint `.pptx` (slide viewer), Markdown (rendered/raw toggle), images, and sandboxed HTML, with `Settings > Interface > iframe Sandbox Allow Same Origin` (off by default) granting previewed HTML same-origin access to Open WebUI. Selecting two files (checkbox or Ctrl-click) and choosing Compare opens a side-by-side diff across formats, later extended with line numbers, inline highlighting, added/removed counts, and a whitespace-ignore toggle. AI-generated files also appear as inline chat cards with preview, download, and open-in-browser buttons plus deep-links to a specific PDF/DOCX page or PPTX slide. Files and folders can be marked Read-only to disable New File , New Folder , Upload , Rename , Delete , and drag operations; other files get an in-browser editor with save support. The browser also supports folder upload preserving subdirectory structure, opens a per-command terminal tab for every command the model runs (alongside the user's own shell tab) for live monitoring and manual takeover, and persists its current folder across chat and terminal switches, auto-refreshing when the AI creates or modifies files.

Allow a previewed HTML file (e.g., an AI-generated report) to make same-origin requests to Open WebUI — useful when the page relies on Open WebUI APIs but carries an understood trust risk.

> 📍 Settings › Interface › iframe Sandbox Allow Same Origin → enable (off by default; applies to ALL embedded iframes, not just terminal file previews)

Route files attached in the chat input directly to the terminal filesystem so the model works with them via terminal tools instead of extracting text — useful for large spreadsheets or binaries.

> 📍 In the terminal connection settings, set `Chat Uploads` to `Filesystem`; attachments then land in the terminal's current working directory and appear in the file browser automatically.

Have AI-generated files appear as inline cards inside the chat reply (with preview, download, and page-level deep-links) rather than opening in the sidebar file browser.

> 📍 Settings › Interface › Terminal File Display → set to `Inline` (default is `Sidebar`)

— Extensive named settings and mechanism; mostly UI toggles, not commands `v0.11.4`

---

02 **Slim Docker image without local ML stack** NEW 90

A slim image variant (`USE_SLIM=true`, published as `ghcr.io/open-webui/open-webui:main-slim`) removes `torch`, `sentence-transformers`, `transformers`, `unstructured`, `ffmpeg`, `pandoc`, and the build toolchain for deployments using hosted model providers. It enforces boot-time validation, refusing to start unless the database is SQLite or PostgreSQL (MySQL, MariaDB, Oracle, and AWS RDS IAM auth require the standard image) and file storage is local (S3, Azure Blob, and Google Cloud Storage require the standard image). It supports `VECTOR_DB=pgvector` with `PGVECTOR_DB_URL` as the only bundled vector store client (others return 503 on first retrieval), routes code interpreter package fetches to `cdn.jsdelivr.net`, removes the `Playwright` web loader and `Transformers` text splitter, and natively extracts plain text for `csv`, `html`, `txt`, `md`, `markdown`, `rst`, and `xml` without any external service — making it compatible with air-gapped deployments. The slim image ships at roughly 175 MB, about 89% smaller than the prior release; the standard image was also reduced by roughly 170 MB by removing a redundant Python copy, unused font sets, and unreferenced packages.

Pull the slim image when you want a lightweight Open WebUI front end with no local ML dependencies, pointed at hosted models.

```
$ docker pull ghcr.io/open-webui/open-webui:main-slim
```

— Names the flag, image tag, env vars and size numbers; docker pull given v0.11.4

03 **Terminal skills discovery and creation** NEW 80

Adds the `/skills:create` chat command to convert a completed chat workflow into a reusable skill written to `.agents/skills` under the connected terminal's root working directory. Skills offered by a connected terminal server now appear in the `$` and `/` menus, the integrations menu, and the skills panel — each marked Terminal — and the model can open them autonomously. Open WebUI automatically reads `AGENTS.md` from the connected terminal's home directory at the start of every turn, and automatic skill discovery lists every available skill by name and description to the model, loading full text only when selected.

— Names the chat command and file path but lacks limits or scope detail v0.11.4

04 **Exa content cap and Staan EU web search provider** NEW 78

Adds `EXA_MAX_CONTENT_LENGTH` (also configurable in admin web search settings) to cap the characters returned per Exa result, and adds Staan as a European Union web search provider configured via `STAAN_API_KEY`, `STAAN_MARKET`, and `STAAN_MAX_SNIPPETS` environment variables or the admin web search settings panel.

Cap Exa search result size to keep context windows manageable in RAG pipelines.

```env
EXA_MAX_CONTENT_LENGTH=2000
```

Route web search through an EU-hosted provider and limit how many snippets are returned.

```env
STAAN_API_KEY=your_key_here
STAAN_MARKET=de-DE
STAAN_MAX_SNIPPETS=5
```

— Named env vars with example values; no default or limit given v0.11.4

05 **Custom file metadata propagation via RAG_SOURCE_METADATA_KEYS** NEW 75

Metadata attached to uploaded files now travels with chunks through retrieval, and the `RAG_SOURCE_METADATA_KEYS` environment variable lets operators specify which custom file metadata fields are passed to the model alongside retrieved source text.

Expose only the source and classification custom metadata fields to the model during retrieval, keeping internal fields out of the prompt.

```env
RAG_SOURCE_METADATA_KEYS=source,classification
```

— Named env var with example usage; propagation mechanism kept brief v0.11.4

06 **Multi-language content and interface localization** NEW

69

Adds per-language names, descriptions, starter prompts, and valve labels for models, tools, skills, functions, banners, and arena entries, editable in a searchable table or importable as a JSON file. Adds an admin panel for replacing interface wording language-by-language, validating that translations don't drop required placeholders. Adds full-text settings search matching each setting's own name and description across all languages, with accent-insensitive matching and permission-aware filtering, plus localized suggested starter prompts on fresh installs shown in the interface language rather than always English.

— Several named surfaces but no exact navigation or file format given v0.11.4

THINNER COVERAGE BELOW

07 **Model background images** NEW

48

A workspace model can carry an uploaded background image shown behind the chat, layered below a folder background and above a personal one, preserved through export and import.

— Describes layering behavior but no config surface or limits v0.11.4

08 **Runtime toggle for email/password sign-in** IMPROVED

47

Adds runtime control to disable the email/password sign-in form from the authentication settings UI, removing the previous requirement to set this before server start.

— Names the UI location but no exact key or path v0.11.4

09 **Diff and patch block rendering in chat replies** NEW

46

Diff and patch code blocks in chat replies are now displayed with file/hunk headings, old/new line numbers, and inline add/remove highlighting, with a toggle back to plain text.

— Clear behavior described, no configuration surface named v0.11.4

10 **Base and Workspace model filters in admin panel** NEW

45

Adds Base models and Workspace models filter options to the admin models page, alongside the existing enabled/disabled/visible/hidden filters.

— Names the exact page and filters but nothing more v0.11.4

11 **Shift-click quick delete for notes and automations** IMPROVED 45

Adds Shift-click quick delete on notes (list and grid views) and automations page rows, collapsing the previous three-step menu-and-confirm flow into a single click.

— Concrete interaction described, minimal further detail v0.11.4

12 **Formatting toggle in notes editor** NEW 45

Adds a Formatting switch in the notes editor menu to disable Markdown rendering and auto-linking, leaving typed or pasted text as literal characters.

— Names the exact menu control, limited additional context v0.11.4

└──▷ BREAKING ON UPGRADE

! Moving a standard-image deployment to the slim image breaks any configuration using MySQL, MariaDB, Oracle, or AWS RDS IAM authentication — only SQLite and PostgreSQL are supported.

! Moving a standard-image deployment to the slim image breaks S3, Azure Blob, and Google Cloud Storage file storage — only local file storage is supported.

! Local Whisper speech-to-text and local Transformers text-to-speech are unavailable in the slim image; deployments relying on them must switch to an external provider or stay on the standard image.

v0.11.4 NOTES ↗ LATEST STABLE

Open WebUI v0.11.4 adds terminal skills, `/skills:create`, `STAAN_*` EU search, `RAG_SOURCE_METADATA_KEYS`, and a side-by-side file differ.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.11.4 https://github.com/open-webui/open-webui.git
# already have the repo? check out this version:
$ git checkout v0.11.4
```

└──▷ USE IT

Cap Exa search result size to keep context windows manageable in RAG pipelines.

```
.env
EXA_MAX_CONTENT_LENGTH=2000
```

Route web search through an EU-hosted provider and limit how many snippets are returned.

.env

```
STAAN_API_KEY=your_key_here
STAAN_MARKET=de-DE
STAAN_MAX_SNIPPETS=5
```

Expose only the `source` and `classification` custom metadata fields to the model during retrieval, keeping internal fields out of the prompt.

.env

```
RAG_SOURCE_METADATA_KEYS=source,classification
```

Adds `/skills:create` chat command to convert a completed chat workflow into a reusable skill written to `.agents/skills` under the connected terminal's root working directory.

Adds `EXA_MAX_CONTENT_LENGTH` environment variable (also configurable in admin web search settings) to cap the number of characters returned per result from Exa web search.

Adds Staan as a European Union web search provider, configured via `STAAN_API_KEY`, `STAAN_MARKET`, and `STAAN_MAX_SNIPPETS` environment variables or the admin web search settings panel.

Adds `RAG_SOURCE_METADATA_KEYS` environment variable letting operators specify which custom file metadata fields are passed to the model alongside retrieved source text.

Adds terminal skills: skills offered by a connected terminal server now appear in the `$` and `/` menus, the integrations menu, and the skills panel — each marked `Terminal` — and the model can open them autonomously.

+18 more

WAS THIS USEFUL? 👍 👎

MLX Serve

Feedback ⓘ v26.9.5 NOTES ↗ RANK

The mlx-serve runtime provides Apple Silicon inference with OpenAI- and Anthropic-compatible APIs, chat, agent mode, and tool calling.

> mlx-serve v26.9.5 adds Qwen-Image-2.1 image generation, a Linux server build via a Vulkan MLX fork, and major batched-MTP throughput gains (up to 119 tok/s), alongside new memory-guard and draft-depth environment controls, Prism Bonsai 2 model support, and a new iOS app; MLX Core.app will be renamed to MLX-Serve.app in the next release.

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

01 **OS memory reserve flag for GPU protection** NEW 85

Adds `--os-reserve-gib` flag (also a toggle in Settings) to reserve OS memory and prevent GPU overrun when multiple long requests are restored from the prefix cache simultaneously.

Cap the OS memory reserve on a 16 GB Mac to prevent kernel panics when multiple long cached requests restore simultaneously.

```
$ mlx-serve --os-reserve-gib 2
```

— Exact flag and runnable command with mechanism explained v26.9.5

02 **Linux server support via Vulkan MLX fork** NEW 80

Adds Linux server support via the mlx-omarchy Vulkan MLX fork, built with `scripts/build-mlx-linux.sh` (Apple-only engines ds4, llama.cpp, and ANE are excluded).

Build and run mlx-serve on Linux using the Vulkan MLX fork.

```
$ bash scripts/build-mlx-linux.sh
```

— Build command given plus explicit list of excluded engines v26.9.5

03 **MTP draft-depth policy environment variable** NEW 75

Adds `MLX_SERVE_MTP_DEPTH_POLICY=legacy` environment variable to restore the previous MTP draft-depth planner behaviour.

Fall back to the previous MTP draft-depth planner if the new auto-depth policy causes regressions in your workload.

```
$ MLX_SERVE_MTP_DEPTH_POLICY=legacy mlx-serve
```

— Named env var with runnable example but planner mechanism unexplained v26.9.5

04 **Qwen-Image-2.1 image generation support** NEW 75

Adds Qwen-Image-2.1 image generation support: text-to-image, image-to-image, and negative-prompt guidance served from `ddalcu/Qwen-Image-2.1-MLX-Serve-8bit` (32 GB Macs) and `-4bit` (16 GB Macs).

— Named model repos and memory tiers but no load command shown v26.9.5

05 **4-stream batched MTP throughput increase** IMPROVED 75

Batches four concurrent MTP streams through a single shared verify forward, raising aggregate throughput on Qwen3.8-27B 4-bit (M4 Max) from 72 to 119 tok/s.

— Clear mechanism and before/after numbers, but automatic and non-actionable v26.9.5

06 **Batched decode KV copy elimination** IMPROVED 75

Eliminates per-step KV copy for batched decode past 1k tokens of context, improving 4-stream throughput at 28k context from 26 to 64 tok/s.

— Precise mechanism and measured gain, no user action required v26.9.5

07 **Prism Bonsai 2 model support** NEW 70

Adds support for Prism Bonsai 2 (`prism-ml/Ternary-Bonsai-2-27B-mlx-2bit`), a Hadamard-rotated 2-bit Qwen3.8-27B model with text and vision, loaded and served like any Qwen 27B.

— Named model id and quantization but only generic load guidance v26.9.5

08 **8-bit KV attention via matmul2d** IMPROVED 65

Routes 8-bit quantized KV attention through Apple's matmul2d tensor op on Qwen and Spark models, matching bf16 KV speed at long context.

— Named op and models, mechanism clear but automatic optimization v26.9.5

THINNER COVERAGE BELOW

09 **MLX Core.app renamed to MLX-Serve.app** BREAKING 50

Next release, MLX Core.app will be renamed to MLX-Serve.app, which will reset app settings and updates.

— Warns of upcoming settings reset but lacks migration steps v26.9.5

10 **New iOS app** NEW 20

Launches a new iOS app powered by this release of MLX-Serve.

— Bare mention with no functional or usage detail v26.9.5

└──▷ ALSO FROM THESE RELEASES

└──▷ BREAKING ON UPGRADE

! Next release, MLX Core.app will be renamed to MLX-Serve.app, which will reset app settings and updates.

v26.9.5 NOTES ↗ LATEST STABLE

mlx-serve v26.9.5 adds image generation, Linux support, 4-stream batched MTP, and an `--os-reserve-gib` memory guard

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v26.9.5 https://github.com/ddalcu/mlx-serve.git
# already have the repo? check out this version:
$ git checkout v26.9.5
```

└──▷ TRY IT

Cap the OS memory reserve on a 16 GB Mac to prevent kernel panics when multiple long cached requests restore simultaneously.

```
$ mlx-serve --os-reserve-gib 2
```

Fall back to the previous MTP draft-depth planner if the new auto-depth policy causes regressions in your workload.

```
$ MLX_SERVE_MTP_DEPTH_POLICY=legacy mlx-serve
```

Build and run mlx-serve on Linux using the Vulkan MLX fork.

```
$ bash scripts/build-mlx-linux.sh
```

Adds `--os-reserve-gib` flag (also a toggle in Settings) to reserve OS memory and prevent GPU overrun when multiple long requests are restored from the prefix cache

simultaneously.

Adds `MLX_SERVE_MTP_DEPTH_POLICY=legacy` environment variable to restore the previous MTP draft-depth planner behaviour.

Adds Linux server support via the mlx-omarchy Vulkan MLX fork, built with `scripts/build-mlx-linux.sh` (Apple-only engines ds4, llama.cpp, and ANE are excluded).

Adds Qwen-Image-2.1 image generation support: text-to-image, image-to-image, and negative-prompt guidance served from `ddalcu/Qwen-Image-2.1-MLX-Serve-8bit` (32 GB Macs) and `-4bit` (16 GB Macs).

Adds support for Prism Bonsai 2 (`prism-ml/Ternary-Bonsai-2-27B-mlx-2bit`), a Hadamard-rotated 2-bit Qwen3.8-27B model with text and vision, loaded and served like any Qwen 27B.

+4 more

└──▷ BREAKING ON UPGRADE

! Next release, MLX Core.app will be renamed to MLX-Serve.app, which will reset app settings and updates.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Pinecone

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗ RANK

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

> Pinecone's Assistant now supports Google's Gemini 3.5 Flash model, pod-to-serverless migration scales to much larger indexes, and Starter plan Assistant usage resets monthly instead of accumulating.

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

01 **Gemini 3.5 Flash model in Assistant** NEW 75

Pinecone Assistant chat requests can now specify `model: "gemini-3.5-flash"` to use Google's Gemini 3.5 Flash model, e.g. via `POST https://api.pinecone.io/assistant/chat` with `{"model": "gemini-3.5-flash", "messages": [...]}`.

Switch an Assistant chat request to the newly supported Gemini 3.5 Flash model.

```
$ curl -X POST https://api.pinecone.io/assistant/chat \
  -H 'Api-Key: <PINECONE_API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gemini-3.5-flash", "messages": [{"role":
"user", "content": "Summarize the attached document."}]}'
```

— Names exact model string and runnable API call changelog-20260922-bccdf269

THINNER COVERAGE BELOW

02 **Higher pod-to-serverless migration limit** IMPROVED 50

Pod-to-serverless migration now supports indexes with up to 500 million records, up from the previous 50 million limit.

— Clear before/after numbers but no command shown changelog-20260922-bccdf269

03 **Monthly reset of Starter Assistant allowances** IMPROVED 30

Pinecone Assistant on the Starter plan now resets included allowances monthly per billing period instead of accumulating as all-time project totals.

— Explains behavior change but no config or numbers changelog-20260922-bccdf269

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Pinecone adds Gemini 3.5 Flash to Assistant, raises pod-to-serverless migration limit to 500M records, and resets Starter Assistant allowances monthly.

└──▷ TRY IT

Switch an Assistant chat request to the newly supported Gemini 3.5 Flash model.

```
$ curl -X POST https://api.pinecone.io/assistant/chat \
  -H 'Api-Key: <PINECONE_API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gemini-3.5-flash", "messages": [{"role": "user",
"content": "Summarize the attached document."}]}'
```

Adds `model: "gemini-3.5-flash"` support for Pinecone Assistant chat requests, enabling use of Google's Gemini 3.5 Flash model.

Pod-to-serverless migration now supports indexes with up to 500 million records, up from 50 million.

Pinecone Assistant on the Starter plan now resets included allowances monthly per billing period instead of accumulating as all-time project totals.

WAS THIS USEFUL? 👍 👎

timgordontg engrim

Feedback ⓘ v1.4.9 NOTES ↗ RANK

The Universal Cross-Model Episodic Memory Standard. Local-first, project-scoped SQLite engine for Google Antigravity, Claude Code, Cursor, Codex CLI, GitHub Copilot CLI, and OpenCode.

> engrim v1.4.9 adds native GitHub Copilot CLI support, wiring command hooks, MCP registration, and lifecycle-based memory capture, bringing the tool's universal harness coverage to six AI coding environments.

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

01 GitHub Copilot CLI setup via `engrim setup --copilot` NEW

90

The new `engrim setup --copilot` command wires native GitHub Copilot CLI integration via command hooks and MCP, registering `engrim serve --mcp` in `~/.copilot/mcp-config.json` with write attribution to `copilot`. It also configures a live status line in `~/.copilot/settings.json` and writes usage notes to `~/.copilot/copilot-instructions.md` as part of the same setup flow.

— Names exact command and config file paths written by setup v1.4.9

02 Copilot CLI lifecycle hooks for memory capture NEW

75

engrim wires `sessionStart`, `userPromptSubmitted`, and `agentStop` lifecycle hooks for Copilot CLI: injecting the memory pack before the initial prompt, logging user prompts to the local flight-recorder, and recording assistant turns via a defensive reader over Copilot's local `events.jsonl` transcript.

— Names hooks and transcript file but no direct user action v1.4.9

THINNER COVERAGE BELOW

03 Universal harness coverage reaches 6 environments IMPROVED

40

With Copilot CLI support added, engrim now covers 6 environments: Antigravity, Claude Code, Cursor, Codex CLI, OpenCode, and GitHub Copilot CLI.

— States the milestone but no new mechanism beyond listed integrations v1.4.9

v1.4.9 NOTES ↗ LATEST STABLE

> engrim v1.4.9 adds native GitHub Copilot CLI support via `engrim setup --copilot`, bringing total harness coverage to 6.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.4.9
https://github.com/timgordontg/engrim.git
# already have the repo? check out this version:
$ git checkout v1.4.9
```

Adds `engrim setup --copilot` to wire native GitHub Copilot CLI integration via command hooks and MCP, registering `engrim serve --mcp` in `~/.copilot/mcp-config.json` with write attribution to `copilot`.

Configures live status line in `~/.copilot/settings.json` and usage notes in `~/.copilot/copilot-instructions.md` as part of Copilot CLI setup.

Wires `sessionStart`, `userPromptSubmitted`, and `agentStop` lifecycle hooks for Copilot CLI: injecting the memory pack before the initial prompt, logging user prompts to the local flight-recorder, and recording assistant turns via a defensive reader over Copilot local `events.jsonl` transcript.

Extends universal harness coverage to 6 environments: Antigravity, Claude Code, Cursor, Codex CLI, OpenCode, and GitHub Copilot CLI.

WAS THIS USEFUL? 👍 👎

Microsoft MarkItDown

Feedback ⓘ v0.1.8 NOTES ↗ RANK

MarkItDown converts documents, spreadsheets, presentations, and other files into Markdown for language-model workflows.

> MarkItDown adds configurable Azure Content Understanding (and Document Intelligence) integration, a new markitdown-ocr plugin for image-heavy PDFs, narrower convert_* APIs for safer untrusted-input handling, short YouTube URL support, and preserved strikethrough formatting.

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

01 **markitdown-ocr plugin for image-heavy PDFs** NEW 72

The `markitdown-ocr` plugin extracts text from image-heavy PDFs without needing extra ML libraries installed, reusing the same LLM client (e.g. OpenAI's `gpt-4o`) already passed via `enable_plugins=True` for image descriptions.

Use the `markitdown-ocr` plugin to pull text out of image-heavy PDFs without installing extra ML libraries — just reuse the same OpenAI client you already pass for image descriptions.

python

```python
from markitdown import MarkItDown
from openai import OpenAI

md = MarkItDown(
    enable_plugins=True,
    llm_client=OpenAI(),
    llm_model="gpt-4o",
)
result = md.convert("document_with_images.pdf")
print(result.markdown)
```

— Runnable example but plugin's internal mechanism is unstated product docs

02 **Short YouTube URL support** NEW 60

Supports short YouTube URLs (e.g. `youtu.be/…`) in addition to full `youtube.com` URLs when converting video links to Markdown.

Convert a short YouTube URL directly — useful in pipelines where video links arrive in the `youtu.be` format.

```
$ markitdown https://youtu.be/dQw4w9WgXcQ > transcript.md
```

— Runnable command shown, minimal added mechanism v0.1.8

THINNER COVERAGE BELOW

03 **Strikethrough formatting preserved in output** IMPROVED

55

Preserves strikethrough formatting from `<strike>` tags, CSS `line-through`, and DOCX `w:dstrike` in Markdown output.

— Names three source formats but no usage example given v0.1.8

04 **Narrowed convert_* APIs for untrusted input** BREAKING

25

Narrows the `convert_*` APIs to reduce risk when handling untrusted input files.

— No detail on which APIs changed or migration steps product docs

v0.1.8 NOTES ↗ LATEST STABLE

MarkItDown v0.1.8 adds `MARKITDOWN_CU_ENDPOINT` and `MARKITDOWN_DOCINTEL_ENDPOINT` env vars, short YouTube URL support, and strikethrough preservation.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.8
https://github.com/microsoft/markitdown.git
# already have the repo? check out this version:
$ git checkout v0.1.8
```

└──▷ TRY IT

Set the Azure Content Understanding endpoint once in the environment so every `--use-cu` invocation picks it up without repeating the flag.

```
$ export MARKITDOWN_CU_ENDPOINT="https://<your-
endpoint>.cognitiveservices.azure.com/"
markitdown path-to-file.pdf --use-cu
```

Convert a short YouTube URL directly — useful in pipelines where video links arrive in the `youtu.be` format.

```
$ markitdown https://youtu.be/dQw4w9WgXcQ > transcript.md
```

> Supports short YouTube URLs (e.g. `youtu.be/…`) in addition to full `youtube.com` URLs.
>
> Preserves strikethrough formatting from `<strike>` tags, CSS `line-through`, and DOCX `w:dstrike` in Markdown output.

WAS THIS USEFUL? 👍 👎

ralforion OrionBelt Ontology Builder

Feedback

v1.27.8 NOTES ↗ RANK

Browser-based ontology workbench for OWL ontologies and SKOS vocabularies. Streamlit + rdflib, no Java, no Protégé.

> OrionBelt refined its entity picker UI across search ranking, multi-select behavior, filter defaults, and accessibility in this release.

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

01 **Entity picker search, selection and accessibility improvements**

IMPROVED 66

All entity pickers (single and multi-select) now run browser-side search that ranks exact-case matches first, then whole-name matches above prefix matches, and prefix matches above mid-word matches. Multi-select pickers preserve selection order (required for property chains) and show picked items as chips with individual × dismiss buttons. SKOS scheme filters and Properties domain filters now clear to `all` instead of offering a separate `All` entry, and picker help text is now exposed as the field's accessible description for screen readers.

— Explains ranking and chip mechanics but no exact UI path given. v1.27.8

v1.27.8 NOTES ↗ LATEST STABLE

OrionBelt v1.27.8 adds case-sensitive ranking to all entity pickers with browser-side search and accessible help text.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.27.8
https://github.com/ralforion/orionbelt-ontology-builder.git
# already have the repo? check out this version:
$ git checkout v1.27.8
```

All entity pickers (single and multi-select) now run browser-side search that ranks exact-case matches first, whole-name matches above prefix matches, and prefix matches above mid-word matches.

Multi-select pickers preserve selection order (required for property chains) and display picked items as chips with individual × dismiss buttons.

SKOS scheme filters and Properties domain filters now clear to `all` instead of offering an `All` entry.

> Picker help text is now exposed as the field's accessible description, making it available to screen readers.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ changelog · Sep 22, 2026

CHANGELOG ↗ RANK

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

> LangSmith's biggest addition this window is a Trajectory/Turns view for inspecting and exporting thread conversations, alongside a new Jev-as-a-judge evaluator from TypeSafe and a shared system sandbox snapshot catalog; dozens of smaller API, Gateway, sandbox, and UI additions round out tracing, evaluation, and deployment surfaces.

└──▷ WHAT SHIPPED · 39 FEATURES most completely described first ? what's the number?

01 System sandbox snapshot catalog NEW

85

Introduces system sandbox snapshots (`system/default:latest` , `system/custom-apps:latest` , `system/code-evaluators:latest`) as a shared read-only catalog, selectable from the same sandbox and Fleet selectors as workspace snapshots. `system/code-evaluators:latest` includes NumPy, pandas, jsonschema, SciPy, and scikit-learn preinstalled so evaluators can use ACE's supported Python packages without installing them.

Create a sandbox from the prebuilt code-evaluator snapshot so your evaluators have NumPy, pandas, SciPy, and scikit-learn available without a manual install step.

> 📍 In the LangSmith console, go to Sandboxes › New Sandbox, select `system/code-evaluators:latest` from the system snapshots catalog, and click Create.

— Snapshot names, packages, and setup steps all given. changelog-20260922-f4610758

02 Trajectory and Turns views for threads NEW

85

Adds a Trajectory view tab on threads that renders the full conversation with human and AI turns, tool calls, and tool results in a scrollable, expand/collapse card layout, alongside a Turns view that shows each thread turn as a card with the root run's inputs and outputs for structural scanning. Supports downloading a thread as a Markdown file via the download button in the Trajectory view, including the full conversation transcript with tool calls and tool results. Adds metadata key support on individual runs to control Trajectory view visibility — runs can be excluded from the Trajectory view by importing and applying the exclusion constant.

Export a thread conversation for offline review or sharing with teammates who don't have LangSmith access.

📍 In the thread panel, open the Trajectory view, then click the download button to save the full conversation — including tool calls and tool results — as a Markdown file.

— Views, export, and exclusion constant all named with example. `product docs`

03 **Jev-as-a-judge evaluator via TypeSafe** NEW 80

Adds `TYPESAFE_API_KEY` provider secret (under Settings > Provider secrets) to connect TypeSafe AI as a model provider, and `jev-latest` as a selectable model under the TypeSafe provider in LLM-as-a-Judge evaluator configuration, supporting noul (yes/no probability), choice, and score question types. Jev-as-a-judge evaluators appear in the Evaluators tab of any tracing project, where each typed question maps to its own feedback key that can be filtered, charted, or used to trigger alerts and automations.

— Secret, model ID, and question types all named. `launch-20260921-c4267686`

04 **Thread stats endpoint** NEW 75

Adds `GET /v2/threads/stats` endpoint to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project — including on self-hosted ClickHouse-backed deployments.

— Full endpoint and returned fields named. `snapshot-20260922`

05 **Thread-trace API pagination and filtering** NEW 70

Adds a `TURN_NUMBER` field when listing thread traces to return each trace's chronological position across pages, including on public shared threads, plus new thread-trace API query parameters for filtering root runs and matching runs anywhere in a trace tree.

— Field and endpoint context named, mechanism explained. `changelog-20260922-f4610758`

06 **Configurable license expiration warning** NEW 70

Adds `LICENSE_EXPIRATION_WARNING_DAYS` environment variable so self-hosted LangSmith operators can configure how many days before license expiration the warning banner appears (default: 7).

Warn the ops team two weeks before a self-hosted LangSmith license expires instead of the default seven days.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

— Env var, default, and example config given. `snapshot-20260922`

07 **LLM Gateway provider tagging** NEW 70

Adds `ls_provider` field on OpenAI-compatible model configurations in the LLM Gateway to populate trace metadata and match custom models to provider-specific pricing.

Tag a custom OpenAI-compatible model with its provider so the LLM Gateway populates trace metadata and applies the right pricing.

```json
{
  "model": "my-custom-gpt4o",
  "base_url": "https://my-endpoint.example.com/v1",
  "ls_provider": "openai"
}
```

— Field name and example config given. snapshot-20260922

---

08 **OAuth authorization session ownership options** NEW 70

OAuth authorization sessions now accept `owner_type` and `owner_id` parameters, allowing a managed credential to be authorized for either a user or an agent, and now also accept workspace owners, enabling one managed OAuth credential to be shared across an entire workspace.

— Both new params named, mechanism explained. snapshot-20260922

---

09 **Portable Gateway prompt manifests** IMPROVED 70

Prompts saved with a LangSmith Gateway model now use a standard `ChatOpenAI` manifest with the unified Gateway URL and a `LANGSMITH_GATEWAY_API_KEY` secret reference, making them portable with standard LangChain SDKs.

— Manifest type and secret name both given. snapshot-20260922

---

10 **Dataset export permission requirement** BREAKING 65

Dataset CSV and JSONL exports now require both dataset read and download permissions; direct API requests without download permission return a forbidden response, so existing API clients with read-only permission will receive a forbidden response on export requests.

— Permission model and failure mode both named. snapshot-20260922

---

11 **Add-to-dataset and annotation queue from thread views** NEW 65

The thread header now includes Add to dataset and Add to annotation queue buttons in every thread view (Messages, Turns, and Details), so a whole thread can be added without selecting a run; the add-to-dataset popup for threads now also offers a New Dataset option that creates an empty dataset and adds the selected threads to it.

– Button and popup labels named, exact UI path. `snapshot-20260922`

12 **Evaluators list API agent filter** NEW 60

Adds `agent_id` filter to the evaluators list API to return only evaluators attached to that agent's environments or tagged datasets.

– Endpoint and filter param named, no usage example. `changelog-20260922-f4610758`

13 **Tracing batch upload timeout status codes** IMPROVED 60

`/runs/multipart` and `/runs/batch` endpoints now return `408 Request Timeout` when a tracing batch upload exceeds the server wait time, and `504` when the timeout is server-side.

– Endpoints and status codes named; behavior explained. `changelog-20260922-f4610758`

14 **Scale-to-zero rollback version guard** BREAKING 60

Scale-to-zero deployments now reject rollbacks to revisions below LangGraph API `0.13.0` or without a known compatible version; the rollback dialog directs you to rebuild the target commit with a supported version.

– Version threshold and dialog guidance both named. `changelog-20260922-f4610758`

THINNER COVERAGE BELOW

15 **Annotation queue item count time-range filters** NEW 58

Adds `min_start_time` and `max_start_time` query parameters to the annotation queue item count endpoint for time-range-scoped counts.

– Endpoint and params named, no example given. `changelog-20260922-f4610758`

16 **SDK-managed Insights report configs** NEW 55

The LangSmith Go and Java SDKs can now save, list, update, and delete Insights report configurations — including recurring schedules — enabling programmatic creation of UI-visible Insights reports.

– SDKs and capability named; no exact call given. `changelog-20260922-f4610758`

17 **Granular usage cost breakdowns** NEW 55

Adds cost estimate breakdowns for individual sandboxes and snapshots in Granular usage, with filtering by workspace, date, and resource type and CSV export.

– Feature named with filters and export format, no numbers. `changelog-20260922-f4610758`

18 **IAM-role ECR access for Sandboxes** IMPROVED 55

Commercial private ECR registries in Sandboxes can now assume an AWS IAM role on demand, enabling repository and tag search without expiring saved authorization tokens.

— Mechanism explained, no named config surface. `snapshot-20260922`

19 **Thread-grouped automation webhooks** NEW 50

Automations grouped by thread can now send webhook payloads containing matched conversations and their runs.

— Named trigger and payload contents; no example. `changelog-20260922-f4610758`

20 **Annotation queue bulk add-to-dataset** NEW 50

Adds completed annotation queue items action to `Add to Dataset`, importing each selected thread as a multi-turn example.

— Action and outcome named, no exact steps. `changelog-20260922-f4610758`

21 **Engine inspects tool result contents** IMPROVED 50

Engine now reads the contents of successful tool results — not only their status — to file issues where a call returned less data than claimed or a value arrived truncated.

— Mechanism explained but no named surface. `changelog-20260922-f4610758`

22 **LLM Gateway fallback timeout handling** IMPROVED 50

LLM Gateway fallback chains can now advance to the next model when a candidate does not return response headers within a configured timeout, with active streams continuing after headers arrive.

— Mechanism described, no named config surface. `snapshot-20260922`

23 **Thread query SSE streaming** NEW 50

Thread query clients can request server-sent events (SSE) to receive query results progressively while preserving cursor pagination.

— Named mechanism (SSE) but no endpoint given. `snapshot-20260922`

24 **Bedrock Playground timeout and retry options** IMPROVED 50

Amazon Bedrock models in the Playground now expose request timeout and max retries options, removing the previous 60-second AWS client default cutoff for long-running agent calls.

— Named options and removed default explained. `snapshot-20260922`

25 **Custom apps list creator filter** NEW 45

Adds filtering of the custom apps list by creator, with the selection preserved in the page URL.

– Thin one-line UI addition. changelog-20260922-f4610758

26 **Managed Deep Agents MCP OAuth discovery fallback** IMPROVED 45

Managed Deep Agents can discover OAuth settings from an MCP server's authentication challenge when well-known metadata lookup fails.

– Mechanism named but no config surface given. changelog-20260922-f4610758

27 **Traces table column customization** IMPROVED 45

The Columns menu on a tracing project's traces and runs table now allows deselecting Status and Name, and dragging them into any order.

– Exact UI path named, thin mechanism. changelog-20260922-f4610758

28 **Cache Hit % project stat** NEW 40

Adds Cache Hit % to the project stats sidebar, surfacing the share of input tokens served from the prompt cache.

– Named metric and location; no mechanism detail. changelog-20260922-f4610758

29 **Persistent LangSmith Chat conversations** IMPROVED 40

LangSmith Chat now persists your conversation as you navigate and reload within a browser tab, starting a new one in a new tab or after a browser restart with earlier conversations accessible from history.

– Behavior described, no named config surface. changelog-20260922-f4610758

30 **Dedicated trajectory batch read service** IMPROVED 40

Trajectory batch reads now run on a dedicated in-cluster gRPC service, so load from trajectory evaluations no longer competes with UI and API traffic.

– Infra change named, no reader action. snapshot-20260922

31 **MCP connector Client ID Metadata Documents** IMPROVED 40

LangSmith MCP connectors now use Client ID Metadata Documents when supported, with dynamic client registration retained as a compatibility fallback.

– Named mechanism, no config surface for reader. snapshot-20260922

32 **Dashboard feedback key suggestions** IMPROVED 35

Dashboard chart feedback key suggestions now include keys from both run-level and thread-level feedback.

– Thin description of suggestion source change. changelog-20260922-f4610758

33 **Local red-team checks against Dev targets** NEW 35

Supports running local production-parity red-team checks against supported Dev targets with their configured trace project and repository context.

— Thin description, no named surface. changelog-20260922-f4610758

34 **Slack mention-mode group DM replies** IMPROVED 35

Newly created Slack apps in mention mode respond to explicit mentions in group DMs and their threads.

— Thin behavioral description, no named surface. changelog-20260922-f4610758

35 **Claude Fable 5.1 availability in Playground** NEW 35

Claude Fable 5.1 is now available in the Anthropic, Bedrock, and Vertex AI model selectors in the Playground.

— One-line model availability note. snapshot-20260922

36 **IssueBench Opus 5 regression benchmark** NEW 35

IssueBench now includes a benchmark task that compares pre- and post-Opus 5 agent traces for cost and trajectory regressions.

— Benchmark task named, no metric detail. snapshot-20260922

37 **Workspace switcher accepts workspace IDs** IMPROVED 30

The workspace switcher now accepts workspace IDs in addition to workspace names.

— Thin one-line UI addition. snapshot-20260922

38 **Managed Deep Agent missing-connection messaging** IMPROVED 30

Managed Deep Agent channel replies now name a missing workspace connection and show how an agent developer can configure it, instead of reporting a generic run failure.

— Behavior described, no named surface. snapshot-20260922

39 **Branded OAuth connection result pages** IMPROVED 30

Agent OAuth connections now show a provider-branded result page after authorization, making successful and failed connection attempts visually distinct.

— UI change described, no named surface. snapshot-20260922

└──▷ **BREAKING ON UPGRADE**

! Scale-to-zero deployments block rollbacks to revisions below LangGraph API `0.13.0` or without a known compatible version; the rollback dialog directs you to rebuild the target commit with a supported version.

! Dataset CSV and JSONL exports now require both dataset read and download permissions; existing API clients that have read-only permission will receive a forbidden response on export requests.

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

LangSmith adds SDK Insights management, `TURN_NUMBER` thread pagination, system sandbox snapshots, and new filtering APIs across tracing and annotation queues.

└──▷ HOW TO FIND IT

Create a sandbox from the prebuilt code-evaluator snapshot so your evaluators have NumPy, pandas, SciPy, and scikit-learn available without a manual install step.

📍 In the LangSmith console, go to Sandboxes › New Sandbox, select `system/code-evaluators:latest` from the system snapshots catalog, and click Create.

Adds `agent_id` filter to the evaluators list API to return only evaluators attached to that agent's environments or tagged datasets.

Adds `min_start_time` and `max_start_time` query parameters to the annotation queue item count endpoint for time-range-scoped counts.

Adds `TURN_NUMBER` field when listing thread traces to return each trace's chronological position across pages, including on public shared threads.

Adds thread-trace API query parameters for filtering root runs and matching runs anywhere in a trace tree.

Adds `/runs/multipart` and `/runs/batch` endpoints now return `408 Request Timeout` when a tracing batch upload exceeds the server wait time, and `504` when the timeout is server-side.

+16 more

└──▷ BREAKING ON UPGRADE

! Scale-to-zero deployments block rollbacks to revisions below LangGraph API `0.13.0` or without a known compatible version; the rollback dialog directs you to rebuild the target commit with a supported version.

snapshot-20260922 NOTES ↗ STABLE

LangSmith adds `/v2` /threads/stats, IAM-role ECR support, Gateway fallback timeouts, and `LICENSE_EXPIRATION_WARNING_DAYS` for self-hosted operators.

└──▷ USE IT

Warn the ops team two weeks before a self-hosted LangSmith license expires instead of the default seven days.

.env

```
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Tag a custom OpenAI-compatible model with its provider so the LLM Gateway populates trace metadata and applies the right pricing.

json

```
{
  "model": "my-custom-gpt4o",
  "base_url": "https://my-endpoint.example.com/v1",
  "ls_provider": "openai"
}
```

Adds `GET /v2/threads/stats` endpoint to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project — including on self-hosted ClickHouse-backed deployments.

Adds `LICENSE_EXPIRATION_WARNING_DAYS` environment variable so self-hosted LangSmith operators can configure how many days before license expiration the warning banner appears (default: 7).

Adds `ls_provider` field on OpenAI-compatible model configurations in the LLM Gateway to populate trace metadata and match custom models to provider-specific pricing.

OAuth authorization sessions now accept `owner_type` and `owner_id` parameters, allowing a managed credential to be authorized for either a user or an agent.

OAuth authorization sessions now accept workspace owners, enabling one managed OAuth credential to be shared across an entire workspace.

+15 more

└─▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read and download permissions; existing API clients that have read-only permission will receive a forbidden response on export requests.

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ @arizeai/phoenix-evals@2.6.0 NOTES ↗

RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix's window centers on a new atomic secrets helper for its client SDK and three new Evals evaluators, alongside Agent Skills support spanning the MCP server and PXI, AI SDK model support in classification evaluators, and server-side trace/session filtering.

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

01 Atomic secrets upsert/delete via PUT /v1/secrets NEW

95

Adds `upsertOrDeleteSecrets` helper to the `secrets` subpath of `@arizeai/phoenix-client` to atomically create, update, or delete ordered key/value-or-null batches via `PUT /v1/secrets`, returning only the affected key names while keeping submitted values out of helper output and error messages.

Atomically provision or rotate a set of API keys — and revoke one — in a single call without exposing values in logs.

javascript

```
import { upsertOrDeleteSecrets } from '@arizeai/phoenix-
client/secrets';

const affected = await upsertOrDeleteSecrets([
  { key: 'OPENAI_API_KEY', value: 'sk-...' },
  { key: 'ANTHROPIC_API_KEY', value: 'ant-...' },
  { key: 'OLD_API_KEY', value: null }
]);

console.log('Affected keys:', affected);
```

— Endpoint, helper name, and runnable code example given `@arizeai/phoenix-client@7.13.0`

02 Custom Agent Skills mounting in MCP and PXI NEW

85

Phoenix MCP server gains a `PHOENIX_SKILLS_PATHS` environment variable accepting comma-separated absolute paths to mount custom Agent Skills directories, making them available in the server's instructions and the Phoenix Assistant's skill picker; each skill directory must include a `SKILL.md` with `name` and `description` frontmatter and

an optional `references/` subdirectory for loadable references. Configured skills are also mounted in MCP and PXI so agents can invoke them directly within those surfaces.

— Names env var and file format but no runnable example `arize-phoenix-v20.15.0`

03 AI SDK evaluation models in classification evaluators NEW

70

Classification evaluators now accept AI SDK evaluation models (e.g., TypeSafe's Jev) as the `model` argument, routing classification through `experimental_evaluate` as a single-choice question rather than `generateObject`, and returning `label` and `score` fields.

— Names mechanism and fields but no code example `@arizeai/phoenix-evals@2.6.0`

04 Filter expressions for getTraces and listSessions NEW

60

Adds a `filter` expression parameter to `getTraces` and `listSessions` in `@arizeai/phoenix-client` for server-side filtering of traces and sessions, requiring Phoenix server >= 20.12.0.

— Names methods and version requirement but no example `@arizeai/phoenix-client@7.12.0`

THINNER COVERAGE BELOW

05 Retrieval relevance, session PII, and completeness evaluators NEW

50

Phoenix Evals adds a retrieval relevance evaluator for scoring how well retrieved documents match a query, a session-level `pii_detection` evaluator that scans across an entire conversation session for personally identifiable information, and a `completeness` evaluator that measures whether a response fully addresses the input.

— Names three evaluators but no config or invocation details `arize-phoenix-evals-v3.9.0`

06 Agent search of GraphQL schema via phoenix-gql NEW

30

Enables agents to search the GraphQL schema from `phoenix-gql`.

— Single-sentence description with no usage detail `arize-phoenix-v20.15.0`

`@arizeai/phoenix-evals@2.6.0` NOTES ↗ LATEST STABLE

Classification evaluators now accept AI SDK evaluation models, routing through `experimental_evaluate` as a single-choice question.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch @arizeai/phoenix-evals@2.6.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout @arizeai/phoenix-evals@2.6.0
```

@arizeai/phoenix-client@7.13.0 NOTES ↗ STABLE

Adds `upsertOrDeleteSecrets` helper to atomically create, update, or delete secrets via `PUT /v1/secrets`

└──▷ GET THIS VERSION ▾

```
$ git clone --branch @arizeai/phoenix-client@7.13.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout @arizeai/phoenix-client@7.13.0
```

└──▷ USE IT

Atomically provision or rotate a set of API keys — and revoke one — in a single call without exposing values in logs.

javascript

```
import { upsertOrDeleteSecrets } from '@arizeai/phoenix-
client/secrets';

const affected = await upsertOrDeleteSecrets([
  { key: 'OPENAI_API_KEY', value: 'sk-...' },
  { key: 'ANTHROPIC_API_KEY', value: 'ant-...' },
  { key: 'OLD_API_KEY', value: null }
]);

console.log('Affected keys:', affected);
```

arize-phoenix-v20.15.0 NOTES ↗ STABLE

Phoenix agents can now mount configured skills in MCP and PXI, and search the GraphQL schema via phoenix-gql.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch arize-phoenix-v20.15.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout arize-phoenix-v20.15.0
```

Mounts configured skills in MCP and PXI so agents can invoke them directly within those surfaces.

Enables agents to search the GraphQL schema from `phoenix-gql`.

`arize-phoenix-evals-v3.9.0` NOTES ↗ STABLE

Phoenix Evals 3.9.0 adds retrieval relevance, session-level PII detection, and completeness evaluators.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch arize-phoenix-evals-v3.9.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout arize-phoenix-evals-v3.9.0
```

Adds a retrieval relevance evaluator for scoring how well retrieved documents match a query.

Adds a session-level `pii_detection` evaluator that scans across an entire conversation session for personally identifiable information.

Adds a `completeness` evaluator that measures whether a response fully addresses the input.

`@arizeai/phoenix-client@7.12.0` NOTES ↗ STABLE

Phoenix client adds `filter` expressions to `getTraces` and `listSessions` for server-side filtering of traces and sessions.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch @arizeai/phoenix-client@7.12.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout @arizeai/phoenix-client@7.12.0
```

> Adds `filter` expression parameter to `getTraces` and `listSessions` (requires Phoenix server >= 20.12.0).

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.41.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse v4.41.0 adds tag-based filtering for prompt events in automations and re-encodes v4 native blocks and event bytes in Rust to speed up ingestion.

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

01 **Rust-encoded v4 native blocks** IMPROVED 38

Langfuse now encodes v4 native blocks and event bytes in Rust, aimed at improving ingestion performance.

— States mechanism and goal but no benchmark or config detail v4.41.0

> v4.41.0 NOTES ↗ LATEST STABLE
>
> Langfuse v4.41.0 adds tag-based filtering for prompt events in automations and Rust-encoded v4 native blocks.
>
> └──▷ **GET THIS VERSION** ▾
>
> ```
> $ git clone --branch v4.41.0
> https://github.com/langfuse/langfuse.git
> # already have the repo? check out this version:
> $ git checkout v4.41.0
> ```
>
> Encodes v4 native blocks and event bytes in Rust for improved ingestion performance.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗

RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust's biggest window addition is Patterns, a scheduled automated investigation that surfaces recurring failure modes, alongside a new Jev integration for tracing and judging decision-model calls, broad SDK auto-instrumentation across OpenAI Agents, LangGraph, Google Discovery Engine, Gemini and Anthropic, and a breaking change to how project vs. organization AI providers select models.

└──▷ WHAT SHIPPED · 22 FEATURES most completely described first ? what's the number?

01 S3 attachment uploader tuning via environment variables NEW

85

Adds `BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE`, `BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS`, `BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES`, and `BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS` environment variables (also available as `BraintrustConfig` options) to tune S3 attachment uploader behavior.

Tune the S3 attachment uploader retry behavior in a high-latency environment to avoid premature failures.

```
$ BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS=30000
BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES=10
BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS=500
python my_eval_script.py
```

— All four env vars named with a runnable tuning command `changelog-20260922-0a2adc66`

02 Selected models for project and organization AI providers

80

BREAKING

Adds a Selected models option for project AI providers on Braintrust-hosted deployments, and model selection for organization providers on both hosted and self-hosted deployments. Project providers now override organization providers only for models they explicitly serve, while organization providers can serve other models using their own credentials and billing accounts — review organization provider model

selections and permissions if you relied on project providers to restrict model access or spending.

— Explains before/after behavior with clear migration guidance `snapshot-20260922` `changelog-20260922-0a2adc66`

03 **OpenAI Agents API tracing in TypeScript and Python SDKs** NEW

80

Adds `startOpenAIAgentsTrace`, `updateOpenAIAgentsTrace`, and `failOpenAIAgentsTrace` functions in the TypeScript SDK (v3.33.0+) for tracing the OpenAI Agents API (`openai.beta.agents.sessions`) from v7+, and adds OpenAI Agents API instrumentation in the Python SDK (v0.40.0+) for beta.agents.sessions.create(stream=True) / `openai.agents.sessions.create`, capturing agent and session metadata, tool call child spans, and token usage.

— Names every function and endpoint across both SDKs `changelog-20260922-0a2adc66`

04 **Removal of `--relabel-overlap` flag from `bt topics config`**

75

BREAKING

Removes the `--relabel-overlap` flag from `bt topics config enable` and `bt topics config set`; it was deprecated and ignored since data plane v2.15.0, and scripts still passing it will fail (see v0.20.0 migration instructions).

— Names exact subcommands and flag with migration pointer `changelog-20260922-0a2adc66`

05 **TypeSafe Jev integration for tracing decision-model calls** NEW

70

Adds TypeSafe Jev integration for tracing structured decision-model calls — context, question criteria, and selected answers — with confidence and per-choice probabilities recorded per judgment (requires Python v0.41.0+ or TypeScript v3.34.0+, data plane v2.15.0+).

— Names versions and captured fields but no runnable snippet `snapshot-20260922`

06 **OpenAI span enrichment and legacy completions tracing** NEW

70

Adds tracing for the legacy OpenAI `/v1/completions` endpoint as a new span type, and adds `prompt_cache_key`, `safety_identifier`, `web_search_options`, `prediction`, and streaming citation `annotations` fields to OpenAI chat completion and responses spans (TypeScript SDK v3.33.0+).

— Names all fields and endpoint but no example call `changelog-20260922-0a2adc66`

07 **`bt observability template` commands** NEW

70

Adds `bt observability template` commands in `bt` v0.19.0+ to export a project's observability configuration as a portable JSON template and apply it to another project.

— Names exact CLI command and its input/output format `snapshot-20260922`

08 **Trace.getSpans() method in TypeScript SDK** NEW 65

Adds Trace.getSpans() to the TypeScript SDK (v3.33.0+) so callers can filter spans by type, name, error state, metadata, and execution time.

— Names method and filter dimensions, no example call `changelog-20260922-0a2adc66`

09 **Project Groups for permission management** NEW 65

Adds Project Groups to let you grant a permission group access to a named collection of projects once, with new projects inheriting access immediately; managed under the Project groups tab.

Grant a permission group access to all current and future projects in a collection without repeating the grant per project.

> 📍 In the console, go to Settings › Project groups › Create group, add your projects, then assign your permission group to the collection once.

— Names UI tab and includes step-by-step usage `product docs`

10 **Per-project breakdown in Usage analytics** NEW 65

Adds per-project breakdown to the Usage analytics section of the billing page, showing logs, scores, and Topics token spend by project, exportable as CSV (requires data plane v2.13.0+).

— Names metrics, export format and version requirement `snapshot-20260922`

11 **Jev as a judge in scorers and classifiers** NEW 60

Adds Jev as a judge in scorers and classifiers, scoring outputs against predefined criteria or labeling logs with a fixed category set using either a built-in model or a TypeSafe key.

— Describes behavior but no configuration example `snapshot-20260922`

12 **Auto-instrumentation for LangGraph Platform SDK** NEW 60

Adds auto-instrumentation for the LangGraph Platform SDK (`@langchain/langgraph-sdk`), tracing RunsClient.wait() and RunsClient.stream() calls (TypeScript SDK v3.33.0+).

— Names package and traced methods, no example `changelog-20260922-0a2adc66`

13 **Require all scores setting in Human Review** NEW 60

Adds `Require all scores` setting in Human Review to enforce that assigned reviewers complete every visible score before a review closes.

— Names the setting and its effect clearly `product docs`

---

14 **`SELECT DISTINCT` on scalar columns in Braintrust SQL** NEW

60

Adds `SELECT DISTINCT` on scalar columns in Braintrust SQL queries to deduplicate results, returning one row per unique combination of values (requires data plane v2.15.0+).

— Names SQL syntax and requirement, no query example `snapshot-20260922`

---

15 **deepseek-v4.1-flash built-in model** NEW 60

Adds `deepseek-v4.1-flash` as a built-in model in playgrounds, prompts, and scorers with multimodal reasoning support; also requestable through the Braintrust Gateway with no AI provider setup required.

— Names model and access surfaces, no setup steps needed `snapshot-20260922`

THINNER COVERAGE BELOW

16 **TypeSafe instrumentation for SDK calls and evaluations** NEW

50

Adds TypeSafe instrumentation for synchronous and asynchronous SDK calls (Python SDK v0.41.0+, TypeScript SDK v3.34.0+) and adds TypeSafe instrumentation and tracing for experimental_evaluate() calls (TypeScript SDK v3.34.0+).

— Names the function but gives no usage detail `changelog-20260922-0a2adc66`

---

17 **Google Discovery Engine instrumentation** NEW 50

Adds Google Discovery Engine instrumentation for answer generation, conversational search, grounding checks, and ranking (Python SDK v0.41.0+).

— Lists covered capabilities but no code example `changelog-20260922-0a2adc66`

---

18 **Multimodal embedding support for Gemini integration** NEW

50

Adds multimodal embedding support to the Gemini integration for `gemini-embedding-2` models, capturing image inputs as Braintrust attachments (TypeScript SDK v3.33.0+).

— Names model and mechanism but no usage snippet `changelog-20260922-0a2adc66`

---

19 **Anthropic context management capture in spans** NEW 50

Adds Anthropic `context_management` config and response `applied_edits` capture together in span metadata (TypeScript SDK v3.33.0+).

— Names fields but no example of use `changelog-20260922-0a2adc66`

20 **Analyze trace view grouping Work sections** NEW 45

Adds an Analyze trace view that groups a trace run into Work sections explaining what the agent did, complementing the Debugger's failure-mode report.

— Describes the view but no navigation path given `snapshot-20260922`

21 **Live topic distribution filtering with autocomplete** NEW 25

Adds live topic distribution filtering via the search bar with autocomplete support.

— Bare description with no mechanism or path detail `changelog-20260922-0a2adc66`

22 **Max reasoning effort for GPT-5.6 Luna** IMPROVED 25

GPT-5.6 Luna now supports Max reasoning effort.

— One-line spec bump with no further detail `changelog-20260922-0a2adc66`

└──▷ BREAKING ON UPGRADE

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; scripts passing this flag will fail (was deprecated and ignored since data plane v2.15.0 — see v0.20.0 migration instructions).

! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers can serve other models using their own credentials and billing accounts — review organization provider model selections and permissions if you relied on project providers to restrict model access or spending.

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials — if you rely on project providers to restrict model access or spending, review organization provider model selections and permissions.

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Braintrust: Adds `BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE`, `BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS`, `BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES`, and `BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS` environment variables (also as `BraintrustConfig` options) to tune S3 attachment uploader behavior.

└──▷ TRY IT

Tune the S3 attachment uploader retry behavior in a high-latency environment to avoid premature failures.

```
$ BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS=30000
BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES=10
BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS=500
python my_eval_script.py
```

Removes the `--relabel-overlap` flag from `bt topics config enable` and `bt topics config set` (deprecated and ignored since data plane v2.15.0; see v0.20.0 migration instructions).

Adds Trace.getSpans() to the TypeScript SDK (v3.33.0+) so callers can filter spans by type, name, error state, metadata, and execution time.

Adds `startOpenAIAgentsTrace`, `updateOpenAIAgentsTrace`, and `failOpenAIAgentsTrace` functions in the TypeScript SDK (v3.33.0+) for tracing the OpenAI Agents API (`openai.beta.agents.sessions`) from v7+.

Adds auto-instrumentation for the LangGraph Platform SDK (`@langchain/langgraph-sdk`), tracing RunsClient.wait() and RunsClient.stream() calls (TypeScript SDK v3.33.0+).

Adds OpenAI Agents API instrumentation in the Python SDK (v0.40.0+) for beta.agents.sessions.create(stream=True) / `openai.agents.sessions.create`, capturing agent and session metadata, tool call child spans, and token usage.

+9 more

└──▷ BREAKING ON UPGRADE

- ! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; scripts passing this flag will fail (was deprecated and ignored since data plane v2.15.0 — see v0.20.0 migration instructions).
- ! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers can serve other models using their own credentials and billing accounts — review organization provider model selections and permissions if you relied on project providers to restrict model access or spending.

snapshot-20260922 NOTES ↗ STABLE

Braintrust adds Patterns auto-investigation, Debugger, Loop automations, Jev typed-output tracing, observability templates, and columnar metric storage.

└──▷ HOW TO FIND IT

Enable Metric columnstore for a high-volume project so the Logs page loads and sorts traces faster without changing query syntax.

📍 In project Settings, find `Metric columnstore` alongside log search optimization and enable it.

Adds `bt observability template` commands in `bt` v0.19.0+ to export a project's observability configuration as a portable JSON template and apply it to another project.

Adds `SELECT DISTINCT` on scalar columns in Braintrust SQL queries to deduplicate results, returning one row per unique combination of values (requires data plane v2.15.0+).

Adds `deepseek-v4.1-flash` as a built-in model in playgrounds, prompts, and scorers with multimodal reasoning support; also requestable through the Braintrust Gateway with no AI provider setup required.

Adds TypeSafe Jev integration for tracing structured decision-model calls — context, question criteria, and selected answers — with confidence and per-choice probabilities recorded per judgment (requires Python v0.41.0+ or TypeScript v3.34.0+, data plane v2.15.0+).

Adds Jev as a judge in scorers and classifiers, scoring outputs against predefined criteria or labeling logs with a fixed category set using either a built-in model or a TypeSafe key.

+5 more

└──▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials — if you rely on project providers to restrict model access or spending, review organization provider model selections and permissions.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.73 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik shipped a full MCP server for agent-driven trace debugging — with hosted OAuth and local stdio modes, CLI configuration, and one-line setup across many AI clients — alongside a new TypeSafe AI integration with automatic span cost tracking.

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

01 **MCP server for agent-driven trace debugging** NEW 98

Opik adds an MCP server that lets coding agents read traces, find failing ones, and verify fixes against real data without requiring the SDK. It runs in two modes: a hosted server at `https://www.comet.com/opik/api/v1/mcp` using HTTP + OAuth (browser sign-in, no API key stored) for Opik Cloud, and a local server via `uvx opik-mcp` over stdio configured with `OPIK_API_KEY` and optional `OPIK_WORKSPACE` in the client's `env` block, with `COMET_URL_OVERRIDE` and `OPIK_URL` supported for self-hosted deployments; MCP OAuth is also supported for self-hosted Opik so agents can authenticate via browser instead of a stored key. Setup is manageable via `uvx opik mcp status` to inspect configured clients and sync state, and `uvx opik mcp configure --ai-client <client> --skills` for non-interactive/CI setup, reading credentials from `~/.opik.config` or `OPIK_API_KEY` / `OPIK_WORKSPACE`; `--ai-client` accepts `claude-code`, `cursor`, `vscode`, `codex`, `opencode`, or `all`, and `--skills` / `--no-skills` control installation of the Opik skill pack. Claude Code can be configured with `claude mcp add --transport http opik-mcp https://www.comet.com/opik/api/v1/mcp` (hosted) or `claude mcp add --transport stdio opik-mcp --env OPIK_API_KEY=<key> --env OPIK_WORKSPACE=<workspace> -- uvx opik-mcp` (local), with equivalent manual config in `~/.claude.json`. Windsurf, Zed, Gemini CLI, Claude Desktop, Goose, Cline, Kiro and others get one-line setup via `npx add-mcp https://www.comet.com/opik/api/v1/mcp --name opik-mcp`, and skill packs for 76+ agents install via `npx skills add comet-ml/opik-skills`.

Check which AI clients are configured and whether they are in sync with your current Opik configuration.

```
$ uvx opik mcp status
```

Run headless MCP setup in CI for Cursor, pulling credentials from environment secrets rather than a terminal prompt.

```
$ uvx opik mcp configure --ai-client cursor --skills
```

Manually configure the local MCP server for Claude Code when you need to connect to a named cloud workspace or self-hosted Opik.

```
$ claude mcp add --transport stdio opik-mcp \
  --env OPIK_API_KEY=<your-key> \
  --env OPIK_WORKSPACE=<your-workspace> \
  -- uvx opik-mcp
```

— Names every flag, endpoint, env var and command across both modes `product docs`

THINNER COVERAGE BELOW

02 **TypeSafe AI integration and cost tracking** NEW 43

Adds TypeSafe AI integration to the SDK, enabling tracing and observability for TypeSafe AI-powered applications, plus backend cost estimation for TypeSafe AI (Jev) spans, surfacing token spend alongside other LLM traces.

— Names the integration and span type but no code or config shown `2.2.73`

> 2.2.73 NOTES ↗ LATEST STABLE
>
> Opik 2.2.73 adds TypeSafe AI integration with automatic cost estimation for Jev spans.
>
> └──▷ GET THIS VERSION ▾
>
> ```
> $ git clone --branch 2.2.73 https://github.com/comet-ml/opik.git
> # already have the repo? check out this version:
> $ git checkout 2.2.73
> ```
>
> Adds TypeSafe AI integration to the SDK, enabling tracing and observability for TypeSafe AI-powered applications.
>
> Adds backend cost estimation for TypeSafe AI (Jev) spans, surfacing token spend alongside other LLM traces.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

agent-safe-pipeline

Feedback ⓘ v0.3.4 NOTES ↗ RANK

Agent Safe Pipeline provides a security-focused framework for building AI agent workflows with policy checks and safety controls.

> Agent Safe Pipeline shipped `govern`, a rewritten Go binary that gates CI/CD workflow steps (GitHub Actions, GitLab CI, Jenkins, and other runners) behind an authorization-and-audit flow, and added Bedrock AgentCore-focused AgentOps integration to its commerce module.

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

01 **govern workflow gate binary** NEW 95

`govern` is a single Go binary — installable via `brew install govern`, a curl installer, or `go install github.com/decionis/agent-safe-pipeline/govern/v2/cmd/govern@v2.1.0` (earlier `@v2.0.0`) — that gates workflow steps across GitHub Actions, GitLab CI, Jenkins, and any runner. It captures execution intent, requests authorization via `enforce-and-bind`, runs commands only on a claimed grant, and finalizes outcomes into the Decision Dossier. It was rewritten in Go as a single static binary for any runner OS/arch, gained a `govern init` command to bootstrap the gate, and added Windows support via a `govern-<version>-windows-x64.zip` archive with PowerShell and cmd shell support.

Install the `govern` gate binary on a CI runner or developer workstation to enforce authorization checks before any workflow step executes.

```
$ go install github.com/decionis/agent-safe-pipeline/govern/v2/cmd/govern@v2.1.0
```

Install the Go-rewritten `govern` binary directly into your Go toolchain to gate workflows on any OS/arch.

```
$ go install github.com/decionis/agent-safe-pipeline/govern/v2/cmd/govern@v2.0.0
```

Bootstrap the `govern` workflow gate in a new environment — runs the bytes pinned to the current commit.

```
$ govern init
```

▶ View command output v0.3.4

— Names install commands, subcommand, and internal grant/dossier mechanism. v0.3.4

v0.3.3

THINNER COVERAGE BELOW

02 **AgentOps integration for Bedrock in commerce module** NEW

45

Adds AgentOps integration for Amazon Bedrock AgentCore via the `commerce-mcp` feature, along with first-use provisioning and durable AgentOps access, and AgentOps historical assessment tools, all in the commerce module.

— Named integrations but no command or mechanism given. v0.3.3

v0.3.4 NOTES ↗ LATEST STABLE

agent-safe-pipeline v0.3.4 adds Windows support for the govern workflow gate, including PowerShell and cmd shells.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.3.4 https://github.com/decionis/agent-safe-pipeline.git
# already have the repo? check out this version:
$ git checkout v0.3.4
```

v0.3.3 NOTES ↗ STABLE

agent-safe-pipeline v0.3.3 ships a Go-rewritten `govern` workflow gate, Amazon Bedrock AgentCore support, and historical AgentOps assessment tools.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.3.3 https://github.com/decionis/agent-safe-pipeline.git
# already have the repo? check out this version:
$ git checkout v0.3.3
```

└──▷ TRY IT

Install the Go-rewritten `govern` binary directly into your Go toolchain to gate workflows on any OS/arch.

```
$ go install github.com/decionis/agent-safe-
pipeline/govern/v2/cmd/govern@v2.0.0
```

Bootstrap the `govern` workflow gate in a new environment — runs the bytes pinned to the current commit.

```
$ govern init
```

Adds `govern init` command as part of the Go-rewritten `govern` workflow gate — a single static binary installable via `go install github.com/decionis/agent-safe-pipeline/govern/v2/cmd/govern@v2.0.0` for any runner OS/arch.

Adds AgentOps integration for Amazon Bedrock AgentCore via the commerce-mcp feature.

Adds first-use provisioning and durable AgentOps access to the commerce module.

Adds AgentOps historical assessment tools to the commerce module.

WAS THIS USEFUL? 👍 👎

FailproofAI

Feedback ⓘ fp-cloud-cli-v0.0.1 NOTES ↗ RANK

Observability and enforcement for AI agent harnesses. Capture every run and runtime reliability with policy enforcement.

> FailproofAI's fp-cloud-cli 0.0.1 introduces a full issue lifecycle toolkit for its CLI, adding won't-fix closing, archive/unarchive, bulk clearing across audits, and closed-state filtering with new timestamp fields.

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

01 **Bulk issue clearing with `fp issues clear`** NEW 95

`fp issues clear (--audit <id> | --all-audits | --everything) [--dry-run]` resolves every open issue in a scope plus their audit findings in one server-side transaction, requiring both `issues:close` and `audits:write` permissions.

Preview how many open issues a bulk clear will affect before committing, then run the actual clear scoped to a single audit.

```
$ fp issues clear --audit a3c9f --dry-run
fp issues clear --audit a3c9f
```

— Full flag set, permission requirements, and transaction semantics given fp-cloud-cli-v0.0.1

02 **Won't-fix closing with `fp issues close`** NEW 85

`fp issues close <id>` ends an issue as won't-fix, distinct from `resolve`: it marks audit findings `dismissed` without writing an org-wide fingerprint suppression, and exits with code 9 if the issue has already ended.

Permanently dismiss a recurring audit finding that you have accepted as won't-fix, without suppressing it org-wide.

```
$ fp issues close 4821
```

— Exact command, distinct semantics from resolve, and exit code given fp-cloud-cli-v0.0.1

03 **Archive and unarchive issues** NEW 65

`fp issues archive <id>` and `fp issues unarchive <id>` take an issue off the board and restore it; both are allowed in any issue state and require no confirmation prompt.

— Commands named with behaviour but no example given `fp-cloud-cli-v0.0.1`

THINNER COVERAGE BELOW

04 **Closed state filter and archive timestamps** NEW 55

`fp issues list` gains `--state closed`, and the `Incident` type gains `closed_at` and `archived_at` fields for tracking lifecycle timestamps.

Pull every closed issue for an audit trail review, using the newly supported `closed` state filter.

```
$ fp issues list --state closed
```

— Named flag and fields but no deeper mechanism explained `fp-cloud-cli-v0.0.1`

`fp-cloud-cli-v0.0.1` NOTES ↗ LATEST STABLE

fp-cloud-cli 0.0.1 adds `fp issues close`, `archive` / `unarchive`, and bulk `clear` commands for issue lifecycle management.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch fp-cloud-cli-v0.0.1
https://github.com/FailproofAI/failproofai.git
# already have the repo? check out this version:
$ git checkout fp-cloud-cli-v0.0.1
```

└──▷ TRY IT

Permanently dismiss a recurring audit finding that you have accepted as won't-fix, without suppressing it org-wide.

```
$ fp issues close 4821
```

Preview how many open issues a bulk clear will affect before committing, then run the actual clear scoped to a single audit.

```
$ fp issues clear --audit a3c9f --dry-run
fp issues clear --audit a3c9f
```

Pull every closed issue for an audit trail review, using the newly supported `closed` state filter.

```
$ fp issues list --state closed
```

Adds `fp issues close <id>` to end an issue as won't-fix, distinct from `resolve`: marks audit findings `dismissed` without writing an org-wide fingerprint suppression, and exits with code 9 if the issue already ended.

Adds `fp issues archive <id>` and `fp issues unarchive <id>` to take an issue off the board and restore it, allowed in any state with no confirmation prompt.

Adds `fp issues clear (--audit <id> | --all-audits | --everything) [--dry-run]` to resolve every open issue in a scope plus their audit findings in one server-side transaction, requiring both `issues:close` and `audits:write` permissions.

Adds `--state closed` to `issues list` and `closed_at` / `archived_at` fields to the `Incident` type.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Google Gemini API

Feedback ⓘ changelog · Sep 22, 2026

CHANGELOG ↗ RANK

Gemini API provides Gemini models for generating text, images, and other multimodal content in applications.

> Gemini API adds Managed Agents with a hosted sandbox that can plan, code, search the web and create files in one request, overhauls the Antigravity agent's built-in file and search tools with a breaking PascalCase rewrite, and ships two GA Live audio-to-audio models.

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

01 Antigravity agent model and built-in tool overhaul

BREAKING

95

Introduces the `antigravity-preview-09-2026` prebuilt agent model as a starting point for agentic tasks, replacing `antigravity-preview-05-2026` (deprecated, shutting down October 5, 2026). Rebuilds the agent's built-in file and search tools with PascalCase parameters: write_file(path, content) (full rewrite) is replaced by write_to_file(TargetFile, CodeContent, Overwrite, Description) for file creation and replace_file_content(TargetFile, StartLine, EndLine, TargetContent, ReplacementContent) for line-range editing; read_file(path, offset, limit) is replaced by view_file(AbsolutePath, StartLine, EndLine, ContentOffset); list_files(path) is replaced by list_dir(DirectoryPath); and new tools find_by_name(SearchDirectory, Pattern, MaxDepth) and grep_search(SearchPath, Query, IsRegex) replace ad-hoc shell commands for search. Agents using `local_environment` or parsing `function_call` steps must update all built-in tool invocations to the new PascalCase parameters and line-range file edits.

Search for hardcoded secrets across a cloned repo using the new built-in `grep_search` tool instead of shelling out — works immediately with the 09-2026 agent.

```json
{
  "agent": "antigravity-preview-09-2026",
  "tool": "grep_search",
  "parameters": {
    "SearchPath": "/workspace/repo",
    "Query": "(password|secret|api_key)\\s*=\\s*['\"].+['\"]",
    "IsRegex": true
  }
}
```

Apply a targeted patch to a specific line range in a source file without overwriting the whole file — use `replace_file_content` to swap out a vulnerable function block.

```json
{
  "agent": "antigravity-preview-09-2026",
  "tool": "replace_file_content",
  "parameters": {
    "TargetFile": "/workspace/repo/auth/login.py",
    "StartLine": 42,
    "EndLine": 58,
    "TargetContent": "def authenticate(user, password):\n
return db.raw_query(f'SELECT * FROM users WHERE password=
{password}')",
    "ReplacementContent": "def authenticate(user, password):\n
return db.query('SELECT * FROM users WHERE password=?',
[password])"
  }
}
```

— Full before/after tool signatures plus two runnable example calls.

changelog-20260922-a6b0fd6c

02 **Managed Agents with hosted sandbox execution** NEW 60

Adds Managed Agents via `CreateAgentInteraction`, giving Gemini a hosted sandbox to plan, write and run code, search the web, and create files in a single request. Adds `CreateAgentInteractionEnvironment` with a `remote` environment option to run agent tasks in a hosted remote sandbox.

— Endpoint and environment option named, but no usage example given. product docs

THINNER COVERAGE BELOW

03 **GA Live audio-to-audio voice models** NEW 55

Releases `gemini-3.8-live` as a GA audio-to-audio model for low-latency voice agent experiences, featuring interleaved reasoning, default asynchronous function calling, and full session client content updates. Releases `gemini-3.8-live-extended-thinking` as a GA audio-to-audio model supporting high background reasoning during live audio interactions via the Live API.

— Model names and capabilities given, but no usage example. changelog-20260922-a6b0fd6c

┗──▷ BREAKING ON UPGRADE

! Agents using `local_environment` or parsing `function_call` steps: built-in tool parameters are now PascalCase instead of snake_case, and file edits use `replace_file_content` with line-range replacements instead of full rewrites via `write_file` — callers must update all built-in tool invocations.

! The `antigravity-preview-05-2026` agent string is deprecated and shuts down on October 5, 2026; callers must migrate to `antigravity-preview-09-2026`.

changelog · Sep 22, 2026 | CHANGELOG ↗ | LATEST | STABLE

Gemini API adds Antigravity Agent 09-2026, two GA Live voice models, and revamped built-in file/search tools with new PascalCase parameters.

┗──▷ USE IT

Search for hardcoded secrets across a cloned repo using the new built-in `grep_search` tool instead of shelling out — works immediately with the 09-2026 agent.

```json
{
  "agent": "antigravity-preview-09-2026",
  "tool": "grep_search",
  "parameters": {
    "SearchPath": "/workspace/repo",
    "Query": "(password|secret|api_key)\\s*=\\s*['\"].+['\"]",
    "IsRegex": true
  }
}
```

Apply a targeted patch to a specific line range in a source file without overwriting the whole file — use `replace_file_content` to swap out a vulnerable function block.

```json
{
  "agent": "antigravity-preview-09-2026",
  "tool": "replace_file_content",
  "parameters": {
    "TargetFile": "/workspace/repo/auth/login.py",
    "StartLine": 42,
    "EndLine": 58,
    "TargetContent": "def authenticate(user, password):\n    return
db.raw_query(f'SELECT * FROM users WHERE password={password}')",
    "ReplacementContent": "def authenticate(user, password):\n
return db.query('SELECT * FROM users WHERE password=?',
[password])"
  }
}
```

Adds `antigravity-preview-09-2026` agent string, replacing `antigravity-preview-05-2026`, with rebuilt built-in file and search tools using PascalCase parameters.

Replaces write_file(path, content) (full rewrite) with write_to_file(TargetFile, CodeContent, Overwrite, Description) for file creation and replace_file_content(TargetFile, StartLine, EndLine, TargetContent, ReplacementContent) for line-range file editing.

Replaces read_file(path, offset, limit) (byte offsets) with view_file(AbsolutePath, StartLine, EndLine, ContentOffset) for file reading.

Replaces list_files(path) with list_dir(DirectoryPath) for directory listing.

Adds find_by_name(SearchDirectory, Pattern, MaxDepth) and grep_search(SearchPath, Query, IsRegex) as new built-in file and code search tools, replacing ad-hoc shell commands.

+2 more

└──▷ BREAKING ON UPGRADE

- ! Agents using `local_environment` or parsing `function_call` steps: built-in tool parameters are now PascalCase instead of snake_case, and file edits use `replace_file_content` with line-range replacements instead of full rewrites via `write_file` — callers must update all built-in tool invocations.
- ! The `antigravity-preview-05-2026` agent string is deprecated and shuts down on October 5, 2026; callers must migrate to `antigravity-preview-09-2026`.

WAS THIS USEFUL? 👍 👎

OpenAI

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗ RANK

OpenAI provides AI models and APIs for building applications that generate and reason with text, images, and audio.

> OpenAI launched a full-featured command-line client for the API covering text responses, image generation, audio, and file uploads, while the Evaluations platform gained support for third-party and self-hosted models and the platform added new API key governance controls.

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

01 **OpenAI CLI global flags and authentication** NEW 90

The new OpenAI CLI adds `OPENAI_API_KEY` and `OPENAI_ADMIN_KEY` environment variables for authentication and `OPENAI_BASE_URL` to target alternative API hosts. It adds a global `--format` flag supporting `auto`, `json`, `jsonl`, `pretty`, `raw`, `yaml`, and `explore` output modes across all commands, a global `--transform` flag to reshape response data via a GJSON path before printing, and a global `--debug` flag to print request/response details (with redacted authorization) to stderr. Installable via `brew install openai/tools/openai` or `go install 'github.com/openai/openai-cli/cmd/openai@latest'` (requires Go 1.25 or later).

— Exact env vars, flags, and install commands given. product docs

02 **openai responses create command family** NEW 90

Adds `openai responses create` for text generation, structured extraction, web search, file understanding, and batch scripting, with a `--text.format` flag to pass a JSON Schema for structured outputs. Supports `web_search` as a hosted tool type in YAML request bodies, and supports YAML heredoc request bodies piped into the command for multiline prompts, nested fields, and shell-assembled inputs.

— Exact command, flag, and tool type named. product docs

03 **Image generation and editing commands** NEW 90

Adds `openai images generate` supporting `--model`, `--prompt`, `--background transparent`, and `--size` flags, with base64 output extraction via `--transform 'data.0.b64_json'`, and `openai images edit` with `--image` and `--prompt` flags for model-driven image editing.

— Exact commands and flags fully enumerated. product docs

04 **Audio speech and transcription commands** NEW 90

Adds `openai audio:speech create` with `--model`, `--voice`, `--input`, `--instructions`, and `--output` flags for local MP3 generation, and `openai`

`audio:transcriptions create` with `--model`, `--file`, `--response-format`, `--timestamp-granularity`, and `--raw-output` flags. Supports `--response-format srt`, `vtt`, `verbose_json`, and `diarized_json` for subtitles, word-level timing, and speaker-labeled diarization.

— Exact commands, flags, and format values enumerated. product docs

05 **Third-party model evaluation via OpenRouter** NEW 85

Enables evaluation of third-party models — Google, Anthropic (hosted on AWS Bedrock), Together, and Fireworks — in the OpenAI Evaluations platform without requiring a separate API key. Access is gated by usage tier 1 or higher and must be enabled by an org admin via `Settings > Organization > General`; inference is served via OpenRouter with monthly spend limits by usage tier: $5 (Tier 1), $25 (Tier 2), $50 (Tier 3), $100 (Tier 4), $200 (Tier 5).

— Exact settings path, tiers, and dollar limits given. product docs

06 **Custom model endpoints for Evaluations** NEW 75

Adds support for custom model endpoints in the OpenAI Evaluations platform, allowing evals against self-hosted models, unsupported providers, or custom inference proxies. Requires an `https://` endpoint compatible with OpenAI's chat completions endpoint and an API key, configurable under Evaluations in Settings on a per-project basis; a Verify button in the setup UI test-calls each configured model slug to surface failures before running evals.

— Names endpoint requirement, config location, and Verify feature. product docs

07 **openai files create command** NEW 70

Adds `openai files create` with `--file` and `--purpose` flags to upload files (e.g. PDFs) and retrieve a file ID for use in response inputs.

— Exact command and flags given, runnable as-is. product docs

08 **API key creation governance controls** NEW 60

Adds API key creation governance controls at the organization and project levels, letting administrators allow only service-account keys, allow only user-owned project keys, or disable all new API key creation entirely. Organization-level restrictions take precedence over project-level settings.

— Describes the three admin options but no exact settings path.

changelog-20260922-fa33f7b9

THINNER COVERAGE BELOW

09 **Expanded file-type support in Responses API** IMPROVED 35

Expands Responses API file-type support to accept document, presentation, spreadsheet, code, and additional text file types beyond what was previously accepted.

— Names file categories but not exact extensions or fields. changelog-20260922-fa33f7b9

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

OpenAI adds API key creation governance controls and expanded file-type support in the Responses API.

Adds API key creation governance controls at the organization and project levels, letting administrators allow only service-account keys, allow only user-owned project keys, or disable all new API key creation entirely; organization restrictions take precedence over project settings.

Expands Responses API file-type support to accept document, presentation, spreadsheet, code, and additional text file types beyond what was previously accepted.

WAS THIS USEFUL?  

Anthropic

Feedback ⓘ changelog · Sep 22, 2026 CHANGELOG ↗ RANK

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

> Anthropic turned on extended thinking by default across most current Claude models and shipped several breaking changes around it (effort-level enforcement, forced-tool-use rejection, model-versioned thinking blocks), while also adding on-demand conversation compaction to the Messages API and a new audit mechanism for thinking-block prefix mismatches.

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

01 On-demand conversation compaction NEW

95

The Messages API adds on-demand conversation compaction via the `compact-2026-09-04` beta header and a top-level `compaction` parameter; the API returns a signed `compaction` block summarizing prior messages, which is sent first on later requests in place of those messages. This lets callers control timing, run compaction in the background, and preserve recent turns verbatim.

Compact a long conversation on demand to stay within context limits while keeping the most recent turns verbatim.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: compact-2026-09-04' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-fable-5-1",
    "max_tokens": 1024,
    "compaction": true,
    "messages": [{"role": "user", "content": "Summarize our
discussion so far."}]
  }'
```

— Full mechanism plus runnable curl example with exact header and param

changelog-20260922-fdd351df

02 Audit logging for thinking-block prefix mismatches NEW

85

Adds `thinking_mismatch_allowed` as a second entry type in the `input_transformations` response field, under the `thinking-binding-controls-`

`2026-08-01` beta header, identifying thinking blocks that failed the prefix check on requests where enforcement is not yet active. This enables audit logging of conversation history edits in production traffic before opting into enforcement via `prefix_mismatch_behavior`.

Log `thinking_mismatch_allowed` entries from production traffic to catch history edits before opting into prefix-check enforcement.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: compact-2026-09-04' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-fable-5-1",
    "max_tokens": 1024,
    "messages": [{"role": "user", "content": "Continue."}]
  }' | jq '.response[] | select(.type ==
"thinking_mismatch_allowed")'
```

— Named field, header and param, with a runnable jq filter example `snapshot-20260922`

03 **Model-versioned thinking blocks** BREAKING 75

Thinking blocks are now versioned by the model that produced them: a block is readable only by that model or a newer one. Claude Fable 5.1 can read blocks from Claude Opus 5, but Claude Opus 5 cannot read blocks from Claude Fable 5.1; the API silently drops unreadable blocks without billing them, so switching to an earlier model drops accumulated reasoning. Later models will enforce append-only conversation history for thinking blocks for all accounts, and integrations not already append-only must be updated now.

— Clear before/after mechanism, but no code example given `product docs`

04 **Extended thinking on by default with effort-level enforcement** 70

BREAKING

Thinking is now on by default, with no configuration required, for `Claude Opus 5`, `Claude Sonnet 5`, `Claude Fable 5.1`, `Claude Mythos 5.1`, `Claude Fable 5`, `Claude Mythos 5`, and `Claude Mythos Preview`. Claude Opus 5 and later models now return a `400` error when requests use effort levels incompatible with thinking enabled, so requests that previously succeeded may now fail.

— Named models and error code but no migration command `product docs`

THINNER COVERAGE BELOW

05 **Progress-update blocks between tool calls** NEW 55

Claude Fable 5.1, Claude Mythos 5.1, and Claude Fable 5 emit a `progress-update` block between tool calls — a short human-readable sentence or two describing what the model just found and what it will do next — returned as its own block.

— Describes the block's content and trigger, no example product docs

06 **Forced tool use rejected on Fable 5.1 and Mythos 5.1** BREAKING 50

Claude Fable 5.1 and Claude Mythos 5.1 now reject forced tool use on every request with a `400` error; adaptive thinking should be used on those models instead.

— Names the error and workaround but no code sample product docs

07 **Interleaved thinking auto-enabled under adaptive thinking** IMPROVED 45

Adaptive thinking now automatically enables interleaved thinking on every model that supports it, with no beta header required; interleaved thinking on manual extended thinking still requires a beta header.

— States the behavior change but no concrete header name given product docs

08 **Local session transcripts for Claude in Chrome** NEW 45

Local session endpoints now return transcripts from Claude in Chrome sessions (`claude_in_chrome`).

— Names the surface but gives no usage detail changelog-20260922-fdd351df

09 **Fable models can refuse to reveal internal reasoning** NEW 30

Claude Fable 5.1 and Claude Fable 5 can refuse requests that attempt to elicit internal reasoning as part of the visible response text.

— One-line behavior note with no mechanism or example product docs

└──▷ BREAKING ON UPGRADE

! Claude Opus 5 and later models return a 400 error when requests use effort levels that are incompatible with thinking enabled — requests that previously succeeded may now fail.

! Claude Fable 5.1 and Claude Mythos 5.1 reject forced tool use on every request with a 400 error.

! Thinking blocks produced by Claude Fable 5.1 or Claude Mythos 5.1 are silently dropped (without error and without billing) if passed to an earlier model — switching down from these models drops the conversation's accumulated reasoning.

! Later models will enforce append-only conversation history for thinking blocks for all accounts; integrations not already append-only must be updated now.

changelog · Sep 22, 2026 CHANGELOG ↗ LATEST STABLE

Anthropic API gains on-demand conversation compaction in beta and expands local session transcripts to Claude in Chrome.

└──▷ TRY IT

Compact a long conversation on demand to stay within context limits while keeping the most recent turns verbatim.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: compact-2026-09-04' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-fable-5-1",
    "max_tokens": 1024,
    "compaction": true,
    "messages": [{"role": "user", "content": "Summarize our
discussion so far."}]
  }'
```

Log `thinking_mismatch_allowed` entries from production traffic to catch history edits before opting into prefix-check enforcement.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: compact-2026-09-04' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-fable-5-1",
    "max_tokens": 1024,
    "messages": [{"role": "user", "content": "Continue."}]
  }' | jq '.response[] | select(.type ==
"thinking_mismatch_allowed")'
```

Adds on-demand conversation compaction to the Messages API via the `compact-2026-09-04` beta header and top-level `compaction` parameter; the API returns a signed `compaction` block summarizing prior messages, which you send first on later requests

in place of those messages, letting you control timing, run compaction in the background, and preserve recent turns verbatim.

Local session endpoints now return transcripts from Claude in Chrome sessions (`claude_in_chrome`).

`snapshot-20260922` NOTES ↗ STABLE

New `thinking_mismatch_allowed` entry type in `input_transformations` lets you audit thinking-block prefix mismatches before enforcing them.

WAS THIS USEFUL? 👍 👎

xAI Grok API

Feedback ⓘ snapshot-20260922 NOTES ↗ RANK

Official SpaceXAI API documentation. Build with Grok: chat, image and video generation, voice, tool calling, and structured outputs.

> Grok API added the grok-4.7 model, a significant upgrade with a 500k context window, image input, tiered reasoning effort, and new pricing tiers.

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

01 **grok-4.7 model launch** NEW 90

Adds `grok-4.7` to the xAI API with a 500k context window, text and image inputs, text-only output, and no text output limit. Supports four `reasoning_effort` levels — `low`, `medium`, `high` (default), and `xhigh`. On the Responses API, `grok-4.7` always returns `reasoning.encrypted_content` even when `include` does not list it, and the model is served on the US regional endpoint. Priced at $2 / $0.50 / $6 per 1M tokens (input / cached input / output) below 200k prompt tokens, and $4 / $1 / $12 above.

— Names model id, params, pricing tiers but no code sample. snapshot-20260922

> snapshot-20260922 NOTES ↗ LATEST STABLE
>
> Grok 4.7 arrives on the xAI API with 500k context, image input, tiered pricing, and four reasoning-effort levels.
>
> Adds `grok-4.7` model to the xAI API with a 500k context window, text and image inputs, text-only output, and no text output limit.
>
> Supports `reasoning_effort` levels `low`, `medium`, `high` (default), and `xhigh` on `grok-4.7`.
>
> On the Responses API, `grok-4.7` always returns `reasoning.encrypted_content` even when `include` does not list it.
>
> Serves `grok-4.7` on the US regional endpoint.
>
> Prices `grok-4.7` at $2 / $0.50 / $6 per 1M tokens (input / cached input / output) below 200k prompt tokens, and $4 / $1 / $12 above.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

▷ Subscribe Feedback on this issue