THE AI TOOLCHAIN

NO. 031

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 19, 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 215 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 30 tools

▾ // THE BRIEF HIDE

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

Three unrelated things stand out. SGLang's v0.5.20 lands sampling masks so RL rollouts replay deterministically, plus a CPU-only duler simulator for tuning without GPUs. LM Studio's Splash engine

vim ? help

cuts local Qwen3 inference time on Apple Silicon. And both Anthropic SDKs now let the tool runner compact context on your trigger, instead of long agent loops dying at the window.

BUILD

Keep a long agent loop alive past the context window on your own terms

Both Anthropic SDKs add an explicit compaction trigger to the agentic tool runner, so a multi-hour tool loop summarises history when you say so rather than hard-failing on overflow or silently dropping turns. PydanticAI's Choices()/Choice API builds structured output schemas from option-sets only known at run time — the case where a static enum in code cannot express the valid answers.

Anthropic Python SDK · Anthropic TypeScript SDK · PydanticAI

DEPLOY

Tune the scheduler and replay RL rollouts without burning GPU hours

SGLang's 713-PR v0.5.20 adds sampling masks so an RL rollout reproduces the exact token choices it made, plus a CPU-only scheduling simulator for capacity and batching experiments that previously needed a real cluster. LM Studio's experimental Metal engine Splash makes Qwen3-series models dramatically faster on Apple Silicon, and Megatron Bridge's automatic GCP NCCL plugin detection removes hand-tuning collectives on supported machine types.

SGLang · LM Studio · Megatron Bridge

BUILD

◆ AI Coding Agents

Augment Code

Feedback ⓘ snapshot-20260919 NOTES ↗ RANK

Your engineers have agents. Your organization doesn't.

> Augment Code expanded its Cosmos platform with eight new source-control, project-management, and monitoring connectors plus CLI service accounts, while the core agent added a configurable sidecar heap limit, approval gating for sensitive-file reads, and a JetBrains IDE version bump.

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

01 Configurable sidecar process heap ceiling

IMPROVED

90

Adds the `-Daugment.sidecar.maxOldSpaceSizeMb` IDE system property to override the sidecar process heap ceiling, which now defaults to 4096 MB, up from Node's roughly 1.7 GB default.

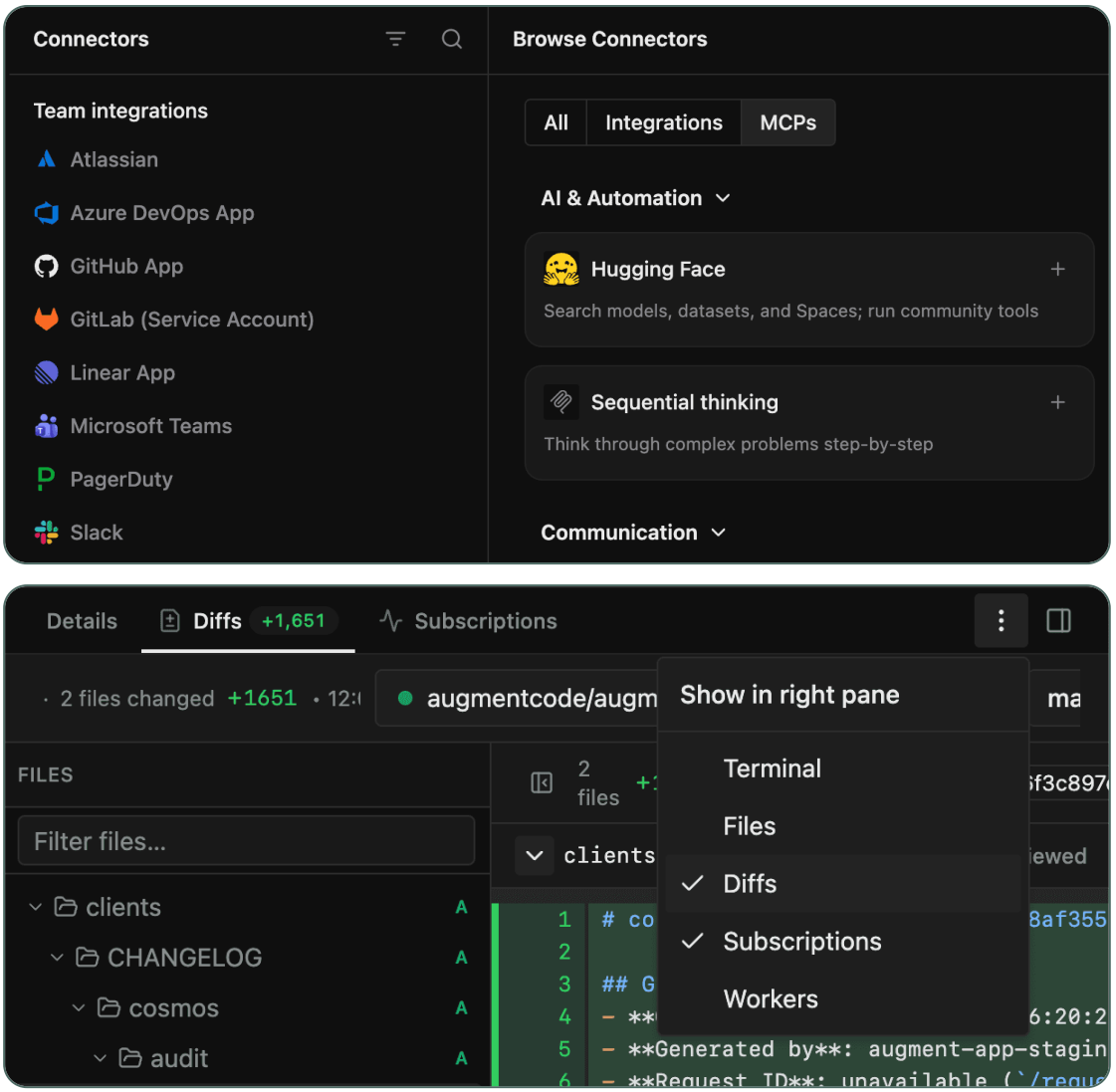
– Exact flag and before/after default values given snapshot-20260919

02 New Cosmos connectors across dev, PM, and monitoring tools

NEW

80

Cosmos gained an Azure DevOps Cloud connector to receive webhook events, access repositories, and automate pull request authoring and code review; a GitHub Enterprise Server (GHES) connector using webhooks, a daemon, and the GitHub CLI to automate pull request reviews on self-hosted infrastructure; a GitLab Cloud connector to receive webhooks, clone repositories, and deploy experts that author and review merge requests; a GitLab Self-Hosted connector using webhooks, a daemon, and the GitLab CLI for merge request workflows on self-hosted infrastructure; an Atlassian Jira Cloud integration with workflow triggers, subscriptions, and user authorization; a ClickUp connector using webhooks and the ClickUp MCP server to receive task events and read workspace objects; a Datadog connector with webhooks and the Datadog MCP server so experts can respond to monitoring events and read logs; and a Twingate connector to configure Cosmos cloud environments for secure access to resources on a remote Linux virtual machine.



— Names each connector's mechanism but gives no setup commands `product docs`

---

03 **IntelliJ plugin requires newer JetBrains IDE** BREAKING

70

The IntelliJ plugin now requires JetBrains IDEs 2025.2 (build 252) or newer; it previously installed on 2024.2 and later. Users on IDEs older than 2025.2 will no longer receive plugin updates.

— Exact version numbers and clear migration consequence given `snapshot-20260919`

---

04 **Approval required before reading sensitive files** NEW

65

The agent now asks for approval before reading sensitive files — `.env` files, private keys, certificates, credential- or secret-named files, and any path excluded by `.augmentignore`.

— Names exact file types and exclusion mechanism, no command shown `snapshot-20260919`

THINNER COVERAGE BELOW

05 **Service accounts for Auggie CLI automation** NEW 45

Adds service accounts for Auggie CLI automation, providing non-human identities with independent API token lifecycles.

— Describes purpose but no setup steps or token details product docs

└──▷ **BREAKING ON UPGRADE**

! The IntelliJ plugin now requires JetBrains IDEs 2025.2 (build 252) or newer; it previously installed on 2024.2 and later. Users on IDEs older than 2025.2 will no longer receive plugin updates.

snapshot-20260919 NOTES ↗ LATEST STABLE

Augment Code adds sensitive-file read approval, proxy-aware MCP servers, workspace trust for hooks, and a 4 GB sidecar heap via `-Daugment.sidecar.maxOldSpaceSizeMb`.

Adds `-Daugment.sidecar.maxOldSpaceSizeMb` IDE system property to override the sidecar process heap ceiling, which now defaults to 4096 MB (up from Node's ~1.7 GB default).

The agent now asks for approval before reading sensitive files — `.env` files, private keys, certificates, credential- or secret-named files, and any path excluded by `.augmentignore`.

└──▷ **BREAKING ON UPGRADE**

! The IntelliJ plugin now requires JetBrains IDEs 2025.2 (build 252) or newer; it previously installed on 2024.2 and later. Users on IDEs older than 2025.2 will no longer receive plugin updates.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ sdk-typescript-v0.1.13

NOTES ↗ RANK

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

> Qwen Code shipped a large cross-surface update: a Playwright-based Browser SDK (`browser-use`) for agent-driven automation, Goal-based token budget tracking and evaluation, a broad Web Shell expansion (Live Voice calling, git branch management, PWA install), and TypeScript SDK additions for multi-modal input, session resume/export, and new config keys — alongside removal of the `read_many_files` and Smart Edit tools and the `active_goal` stream event.

└──▷ **WHAT SHIPPED · 34 FEATURES** most completely described first ? what's the number?

01 **New SDK configuration keys** NEW 78

Adds `settings.env` for configuring environment variables directly in settings without modifying shell profiles, `extra_body` to pass provider-specific parameters to OpenAI-compatible providers, and `contextWindowSize` to control the model context window size.

— Names config keys but no usage example shown sdk-typescript-v0.1.13

02 **Session resume, continue, and export in TypeScript SDK** NEW 78

Adds `resume` and `continue` options plus extended `authType` support to the TypeScript SDK, support for resuming sessions by session ID via the `query` interface, and an `export` command for session history with markdown and HTML output formats.

— Names surfaces but no runnable example given sdk-typescript-v0.1.13

03 **Named workflow execution controls** NEW 78

Enables running saved workflows by name and pins workflow grants to script content, and adds a name-only lock on the Workflow tool — set via `workflowNameOnly: true` — that restricts it to named, registered workflows, blocking ad-hoc script execution.

Restrict the Workflow tool so the model can only invoke named, registered workflows — blocking ad-hoc script execution.

```yaml
tools:
  workflowNameOnly: true
```

— Named config key with a runnable example desktop-v0.24.1 v0.24.1

04 **Web Shell feature expansion** NEW 73

Adds footnote previews and per-turn source citations, file previews from expanded tool details, and the ability to connect to a selected remote daemon. Adds admission as a browser Live Host with Live Voice calling, PWA installability and an Android development shell, git remote management from the workspace branch picker, a commit history lane graph with search across all branches, folding of oversized pastes into an attachment card, and support for host settings item exclusions.

— Many named UI surfaces, no exact navigation path desktop-v0.24.1 v0.24.1

05 **Playwright Browser SDK and Chrome relay** NEW 68

Adds a Playwright-based Browser SDK (`browser-use`) enabling agent-driven browser automation, plus a Chrome Native Messaging relay for the `browser-use` integration that bridges the agent to an installed Chrome extension.

— Names SDK and relay, no setup commands v0.24.1

06 **Hooks listing and event metadata** IMPROVED 62

Adds `enabled` and `disabledReason` fields to the hooks listing, exports hook event display metadata and pure timeout helpers from core, and identifies hook invocations in progress events and announces the message bus.

— Names fields and exports, no example call v0.24.1

07 **read_many_files tool removed for readManyFiles utility** 60

BREAKING

Removes the `read_many_files` tool, replacing it with a `readManyFiles` utility used via user `@ -commands` .

— Names old and new surface plus usage path sdk-typescript-v0.1.13

08 **Goal state exposure and evaluation** NEW 60

Writes Goal state records into session exports from the CLI, shows a Goal's turn and active-time ceilings on the Web Shell status strip, adds per-turn satisfied / not-satisfied marks in the Web Shell, and judges terminal proposals from the transcript tail for Goal evaluation.

— Several named surfaces, no exact command shown v0.24.1

THINNER COVERAGE BELOW

09 **Goal budget tracking and telemetry** NEW 55

Adds Goal lifecycle telemetry events and metrics, a turn and active-time budget display in the Goal UI, and bills foreground subagent and verifier token usage against the Goal budget.

— Describes mechanism, no config path given desktop-v0.24.1

10 **ACP child admission control in serve** NEW 53

Adds budget-based ACP child admission in `serve`, controlling how many child agents are admitted based on token budget, and idle ACP child reclamation so slots are freed when admission is full.

— Explains mechanism, no config example desktop-v0.24.1

11 **Subagent tool allowlists and container execution** NEW 52

Lets agent() narrow a subagent to an explicit tool allowlist, limiting which tools a spawned subagent may call, and adds container execution support for subagents.

— Names function and mechanism, no example v0.24.1

12 **Workflow run reliability and tracing** IMPROVED 51

Preserves source references across workflow runs and resumes, correlates agent dispatches and nested workflow calls for tracing, adds workflow size guidelines, a structure preview, and a large-run warning UI, and records a checkpoint for every workflow run so restarts surface interrupted runs.

— Several thin mechanisms grouped, no command desktop-v0.24.1 v0.24.1

13 **Model-invoked extension workflows via whenToUse** NEW 50

Enables extension workflows that declare `whenToUse` to be invoked directly by the model.

— Names field, brief mechanism given desktop-v0.24.1

14 **Per-model OpenAI wire API selection** NEW 48

Adds model-selectable OpenAI wire API per model, allowing different wire protocols to be chosen per model configuration.

— Explains behaviour, no config snippet `desktop-v0.24.1` `v0.24.1`

15 **Debug output routed through debugLogger** IMPROVED 46

Adds a debug mode output refactor that routes console calls to a logfile-first `debugLogger`.

— Names debugLogger, no config example `sdk-typescript-v0.1.13`

16 **FORK_MODE support in ProcessTransport** NEW 45

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

— Named flag and transport, no usage steps `sdk-typescript-v0.1.13`

17 **active_goal stream event removed** BREAKING 45

The `active_goal` stream event is no longer emitted; any client or integration that listens for it will stop receiving it.

— Names exact event, no migration path given `v0.24.1`

18 **DingTalk channel integration** NEW 44

Adds DingTalk channel support with shared output modes and sender identity in direct messages.

— Names channel and modes, no setup steps `desktop-v0.24.1`

19 **Configurable Mem0 providers** NEW 41

Supports configurable Mem0 providers for external-context memory.

— Names Mem0 integration, no config detail `v0.24.1`

20 **Multi-modal input support** NEW 35

Adds multi-modal input support (image, PDF, audio) across all content generators.

— Brief description, no mechanism detail `sdk-typescript-v0.1.13`

21 **Coding Plan authentication mode** NEW 35

Adds Coding Plan authentication mode with a unified `AuthDialog`.

— Names dialog component, no setup steps `sdk-typescript-v0.1.13`

22 **Paste UI improvements** IMPROVED 35

Adds a large paste placeholder and fixes enter-submit on macOS in the `paste` UI.

— Thin UI fixes only `sdk-typescript-v0.1.13`

23 **Removed Smart Edit tool and ClearcutLogger** BREAKING 35

Removes the Smart Edit tool and `ClearcutLogger`.

— Named removals, no migration guidance sdk-typescript-v0.1.13

24 **Model selection via ACP in VS Code** NEW 35

Adds model selection through ACP in the VS Code IDE companion.

— Names integration, no navigation steps sdk-typescript-v0.1.13

25 **Omni multimodal pipeline in agent runtime** NEW 35

Integrates a multimodal media pipeline (`omni`) into the main agent runtime.

— Names pipeline only, no scope detail desktop-v0.24.1

26 **Summary projection in daemon prompt/replay APIs** NEW 35

Adds summary projection support for prompt and replay APIs in the daemon.

— Names APIs but no endpoint detail desktop-v0.24.1

27 **Agent Skills stability and symlink support** IMPROVED 32

Promotes Agent Skills from experimental to stable and adds symlink support for the skill manager.

— Status change plus a thin addition sdk-typescript-v0.1.13

28 **MCP tool progress updates** NEW 32

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

— Names surfaces, no mechanism explained sdk-typescript-v0.1.13

29 **Local extension skill enumeration via daemon** NEW 29

Adds local enumeration of installed extension skills via the daemon.

— Bare capability, no usage path desktop-v0.24.1

30 **System prompt assembled from resident tool set** IMPROVED 27

Assembles the system prompt from the resident tool set.

— Bare mechanism statement only v0.24.1

31 **Zed extension for Qwen Code** NEW 25

Adds a Zed extension for the Qwen Code agent server.

— Names integration only, no setup detail `sdk-typescript-v0.1.13`

32 **Concurrent batch runner for CLI** NEW 23

Adds a concurrent runner for batch CLI execution.

— One-line description, no command shown `sdk-typescript-v0.1.13`

33 **Cross-session messaging enabled by default** IMPROVED 23

Enables cross-session messaging on by default.

— One-line toggle, no scope explained `desktop-v0.24.1`

34 **Experimental LSP support** NEW 21

Adds experimental LSP support for code intelligence.

— Bare mention, no scope given `sdk-typescript-v0.1.13`

└──▷ **BREAKING ON UPGRADE**

! The `read_many_files` tool is removed; use the `readManyFiles` utility via user `@ -` `commands` instead.

! The Smart Edit tool and `ClearcutLogger` are removed.

! The `active_goal` stream event is no longer emitted; any client or integration that listens for it will stop receiving it.

`sdk-typescript-v0.1.13` NOTES ↗ LATEST STABLE

Qwen Code SDK v0.1.13 adds multi-modal input, session resume, concurrent batch runner, settings.env, session export, LSP, and Zed extension

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch sdk-typescript-v0.1.13
https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout sdk-typescript-v0.1.13
```

Adds `settings.env` field for configuring environment variables directly in settings without modifying shell profiles.

Adds `extra_body` support for OpenAI-compatible providers to pass provider-specific parameters.

Adds `contextWindowSize` configuration key to control the model context window

size.

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

Adds support for resuming sessions by session ID via the `query` interface.

+15 more

└──▷ **BREAKING ON UPGRADE**

! The `read_many_files` tool is removed; use the `readManyFiles` utility via user `@ -commands` instead.

! The Smart Edit tool and `ClearcutLogger` are removed.

desktop-v0.24.1 NOTES ↗ STABLE

Qwen Code: Adds support for external reasoning profiles and defaults in `core`, letting operators supply and pin custom reasoning configurations.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch desktop-v0.24.1 https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout desktop-v0.24.1
```

└──▷ **USE IT**

Restrict the Workflow tool so the model can only invoke named, registered workflows — blocking ad-hoc script execution.

yaml

```
tools:
  workflowNameOnly: true
```

Adds a `+Nk` directive in workflows to share a per-turn token budget across all steps in a turn.

Adds budget-based ACP child admission in `serve`, controlling how many child agents are admitted based on token budget.

Adds idle ACP child reclamation in `serve` so slots are freed when admission is full.

Adds the ability to start workflow runs with `args`, `sourceRef`, and a caller-supplied script via the serve API.

Adds model-selectable OpenAI wire API per model, allowing different wire protocols to be chosen per model configuration.

+16 more

v0.24.1 NOTES ↗ STABLE

Qwen Code v0.24.1 adds Playwright Browser SDK, Chrome Native Messaging relay, configurable Mem0 providers, subagent tool allowlists, and Live Voice in the browser.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.24.1 https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout v0.24.1
```

Adds Playwright-based Browser SDK (`browser-use`) enabling agent-driven browser automation.

Adds Chrome Native Messaging relay for the `browser-use` integration, bridging the agent to an installed Chrome extension.

Lets agent() narrow a subagent to an explicit tool allowlist, limiting which tools a spawned subagent may call.

Adds a name-only lock on the Workflow tool that restricts it to named workflows only.

Supports configurable Mem0 providers for external-context memory.

+18 more

└──▷ BREAKING ON UPGRADE

! The `active_goal` stream event is no longer emitted; any client or integration that listens for it will stop receiving it.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.278 NOTES ↗

CODE ↗ RANK

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

> Claude Code defaults auto mode to a server-side classifier to cut overhead charges, adds AGENTS.md project-instruction support and new gateway proxy/header controls, ships a batch of VSCode extension and web UI additions, and removes the deprecated TaskOutput tool.

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

01 Server-side auto mode classifier by default IMPROVED

90

Auto mode now defaults to a server-side classifier for Claude API, Enterprise, Bedrock, Vertex, Foundry, and gateway users, eliminating classifier overhead charges. A new `Auto mode server` row in `/status` shows whether the current session's classifier runs server-side, and setting `CLAUDE_CODE_AUTO_MODE_SERVER=0` opts back into the previous client-side behaviour on Bedrock, Vertex, Foundry, and gateways.

Opt out of the server-side auto mode classifier on Bedrock, Vertex, Foundry, or a gateway when you need the previous client-side behaviour.

```
$ CLAUDE_CODE_AUTO_MODE_SERVER=0 claude
```

— Names platforms, status field and exact opt-out variable. v2.1.278

02 AGENTS.md project instructions support NEW

90

Claude Code reads `AGENTS.md` for project instructions when no `CLAUDE.md` is present. The `instructionFiles` option (values `claude-md`, `claude-md-or-agents-md`, `claude-md-and-agents-md`, `managed-only`) controls this behavior and is configurable under `Project instructions` in `/config`; not yet available on Bedrock, Vertex, or Foundry.

Allow a project that uses AGENTS.md (e.g. an OpenAI-compatible repo) to supply Claude Code with project instructions without renaming the file to CLAUDE.md.

```yaml
# Place project instructions in AGENTS.md at the repo root;
# Claude Code reads it automatically when no CLAUDE.md is
present.
# To read BOTH files, set instructionFiles in /config > Project
instructions:
# instructionFiles: claude-md-and-agents-md
```

— Names config key, all option values, and exact UI path. v2.1.277

03 **Gateway upstream proxy and header configuration** NEW

85

Claude apps gateways gain two upstream configuration options: `CLAUDE_GATEWAY_PROXY_IS_EGRESS_BOUNDARY=1` routes every outbound request through a forward proxy by hostname instead of resolving it locally, for gateways whose only egress is that proxy; an optional `headers:` map on gateway upstreams sends static headers to a proxy sitting in front of a provider.

Route all outbound Claude gateway traffic through a corporate forward proxy that is the sole egress point, without local DNS resolution.

```
$ CLAUDE_GATEWAY_PROXY_IS_EGRESS_BOUNDARY=1 claude
```

— Exact env var and config key with runnable example given. v2.1.277

04 **TaskOutput tool removed** BREAKING

75

The `taskOutputMaxChars` setting and `TASK_MAX_OUTPUT_LENGTH` environment variable no longer have any effect: the deprecated TaskOutput tool has been removed, and Claude now reads background task output with the Read tool instead.

— Names the exact setting, env var and replacement mechanism. v2.1.277

05 **Background task visibility in /tasks panel** NEW

70

The `/tasks` panel now shows a status line indicating when a background task has finished and its update is waiting. In the VSCode extension, background shells and other running tasks appear in the agent map with a Stop button each, and a typed `/tasks` command opens the map.

— Names the command and UI element but no config surface. v2.1.277

06 **VSCode extension panel UI additions** NEW

60

The VSCode extension panel adds a Sign out row (with a typed `/logout` command), a Copy response button on responses (with a typed `/copy` command), a one-time notice when inactive sessions are auto-archived plus an `Unarchive all` action on the Archived sessions group, and session cost and token usage shown in the Account & usage dialog and session manager for plans where limits do not apply (Vertex, Bedrock, Foundry, API key).

— Names each UI element and command but limited mechanism. v2.1.277

THINNER COVERAGE BELOW

07 Environment picker sharing on Claude Code on the web NEW

55

Claude Code on the web adds Personal and Organization sections to the environment picker for Team and Enterprise plans, and lets admins share a personal environment with the organization.

— Describes scope and plans but only a UI path to start. v2.1.277

└──▷ BREAKING ON UPGRADE

! The `taskOutputMaxChars` setting and `TASK_MAX_OUTPUT_LENGTH` environment variable no longer have any effect; the deprecated TaskOutput tool has been removed and Claude now reads background task output with the Read tool instead.

v2.1.278 NOTES ↗ CODE ↗ LATEST STABLE

Claude Code v2.1.278 defaults auto mode to a server-side classifier and surfaces its status in `/status`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.1.278 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.278
```

└──▷ TRY IT

Opt out of the server-side auto mode classifier on Bedrock, Vertex, Foundry, or a gateway when you need the previous client-side behaviour.

```
$ CLAUDE_CODE_AUTO_MODE_SERVER=0 claude
```

Adds `Auto mode server` row to `/status` showing whether the current session's auto mode classifier runs on the server.

Switches auto mode to default to the server-side classifier for Claude API, Enterprise, Bedrock, Vertex, Foundry, and gateway users — eliminating classifier overhead charges; set `CLAUDE_CODE_AUTO_MODE_SERVER=0` to opt out on Bedrock, Vertex, Foundry, and gateways.

v2.1.277 NOTES ↗ CODE ↗ STABLE

Claude Code v2.1.277 adds AGENTS.md support, gateway egress-proxy env var, and static upstream headers for Claude apps gateways.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.1.277 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.277
```

└──▷ **TRY IT**

Route all outbound Claude gateway traffic through a corporate forward proxy that is the sole egress point, without local DNS resolution.

```
$ CLAUDE_GATEWAY_PROXY_IS_EGRESS_BOUNDARY=1 claude
```

Allow a project that uses AGENTS.md (e.g. an OpenAI-compatible repo) to supply Claude Code with project instructions without renaming the file to CLAUDE.md.

yaml

```
# Place project instructions in AGENTS.md at the repo root;
# Claude Code reads it automatically when no CLAUDE.md is present.
# To read BOTH files, set instructionFiles in /config > Project instructions:
# instructionFiles: claude-md-and-agents-md
```

Adds `CLAUDE_GATEWAY_PROXY_IS_EGRESS_BOUNDARY=1` environment variable for Claude apps gateways whose only egress is a forward proxy, routing every outbound request through the proxy by hostname instead of resolving it locally.

Adds an optional `headers:` map on Claude apps gateway upstreams to send static headers to a proxy in front of a provider.

Adds AGENTS.md support: in a project with no CLAUDE.md, Claude Code reads AGENTS.md instead; the `instructionFiles` option (values: `claude-md`, `claude-md-or-agents-md`, `claude-md-and-agents-md`, `managed-only`) controls the

behavior and is configurable under `Project instructions` in `/config` (not yet on Bedrock, Vertex, or Foundry).

Adds a status line in the `/tasks` panel indicating when a background task has finished and its update is waiting.

[VSCode] Adds a Sign out row to the panel menu, with `/logout` in the typed command menu.

+5 more

└─▷ **BREAKING ON UPGRADE**

! The `taskOutputMaxChars` setting and `TASK_MAX_OUTPUT_LENGTH` environment variable no longer have any effect; the deprecated TaskOutput tool has been removed and Claude now reads background task output with the Read tool instead.

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.156.0-alpha.5 NOTES ↗

CODE ↗ RANK

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

> Codex CLI's alpha release adds six bundled TUI themes, composes gateway OAuth credentials with primary provider auth (including shared ChatGPT cookies across HTTP and WebSocket transports), and tightens macOS sandboxing while allowing `/review` during MCP startup.

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

01 **Gateway OAuth composition and cookie sharing across transports**

IMPROVED 55

Gateway OAuth credentials are now composed with primary provider authentication, including setting WebSocket handshake headers, and ChatGPT cookies are shared between the HTTP and WebSocket transports.

– Names transports and header mechanism but no config surface rust-v0.156.0-alpha.5

02 **Blocks mutating fcntls in restricted macOS Seatbelt policies**

50

BREAKING

Restricted macOS Seatbelt sandbox policies now block mutating `fcntl` calls.

– Names exact syscall and sandbox mechanism, no migration note rust-v0.156.0-alpha.5

03 **`/review` usable during MCP startup** NEW 48

The `/review` command can now be invoked while MCP is still starting up.

– Names exact command; no further mechanism given rust-v0.156.0-alpha.5

04 **Unified TUI exploration command history** IMPROVED 43

TUI exploration commands are kept grouped across reasoning summaries and nonzero exits, in both live sessions and replayed history.

– Describes behavior scope but no user-facing control named rust-v0.156.0-alpha.5

05 **Six bundled TUI themes with theme-aware colors** NEW

40

The terminal UI now ships six bundled themes, with accent colors that adapt to the selected theme.

– Names count of themes but no selection command shown rust-v0.156.0-alpha.5

06 **Unified tool output previews with three-row limit** IMPROVED 38

TUI tool output previews are now unified and capped to a three-row limit.

– Gives concrete limit but no configuration option rust-v0.156.0-alpha.5

07 **Worktree sessions reuse existing local daemon** NEW 35

Worktree sessions can now use an existing local daemon rather than starting a new one.

– States capability but no command or flag given rust-v0.156.0-alpha.5

08 **Approved escalation with environment-owned network policies** NEW 33

Escalation can now be approved when network policies are environment-owned.

– Describes policy condition, no config key or command shown rust-v0.156.0-alpha.5

09 **Catalog model display names in TUI** IMPROVED 23

The TUI now uses catalog model display names throughout, instead of raw identifiers.

– Bare statement with no further detail rust-v0.156.0-alpha.5

10 **Configuration and feature diagnostics in reports** NEW 22

Report metadata now includes configuration and feature diagnostics.

– Only a bare description, no fields or format named rust-v0.156.0-alpha.5

rust-v0.156.0-alpha.5 NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI adds six bundled TUI themes, gateway OAuth composition, and `/review` during MCP startup

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rust-v0.156.0-alpha.5
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.156.0-alpha.5
```

Adds six bundled TUI themes with theme-aware accent colors to the terminal UI.

> Allows the `/review` command to be used during MCP startup.
>
> Adds configuration and feature diagnostics to report metadata.
>
> Composes gateway OAuth credentials with primary provider authentication, including WebSocket handshake headers.
>
> Shares ChatGPT cookies between HTTP and WebSocket transports.
>
> +6 more

WAS THIS USEFUL?  

Graphify Labs Graphify

Feedback ⓘ v0.9.64 NOTES ↗

RANK

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

> Graphify v0.9.64 adds typed, secret-redacted indexing of Terraform resource attributes, extracts inline JavaScript from PHP script blocks, and extends cross-file resolution to Rust generics, Kotlin receivers, and C++ static calls.

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

01 Terraform attribute indexing with secret redaction NEW

85

Graphify indexes Terraform block attributes (`ami` , `instance_type` , `cidr_block` , tags, nested blocks) as queryable, typed (bool/number/list/map) properties on resource/data/module nodes in `graph.json` . Values whose keys match secret patterns (`password` , `*secret*` , `*token*` , `*_key` , connection strings) are automatically redacted before reaching the graph or the model.

After mapping a Terraform repo, query the graph to find all resources whose `cidr_block` attribute is set — without any hardcoded credentials leaking into the output.

```
$ graphify query "which aws_instance resources have instance_type
set to t3.micro"
```

— Names exact attributes, node types, output file, and redaction patterns. v0.9.64

THINNER COVERAGE BELOW

02 Inline JavaScript extraction from PHP script blocks NEW

53

Extracts JavaScript inside inline `<script>` blocks of PHP files as JS (functions and calls) under the PHP file node, with source lines mapped back to real file positions, mirroring the existing Vue/Svelte embedded-script handling.

— Explains mechanism and line-mapping but no runnable example given. v0.9.64

03 **Cross-file resolution for Rust, Kotlin, C++** IMPROVED

28

Adds cross-file resolution for Rust generics, Kotlin receivers, and C++ static calls, per the release summary.

— Only named in the summary line, no mechanism or example given. v0.9.64

v0.9.64 NOTES ↗ LATEST STABLE

Graphify v0.9.64 adds Terraform attribute indexing with secret redaction, inline JS extraction from PHP, and cross-file resolution for Rust generics, Kotlin receivers, and C++ static calls.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.64 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.64
```

└──▷ TRY IT

After mapping a Terraform repo, query the graph to find all resources whose `cidr_block` attribute is set — without any hardcoded credentials leaking into the output.

```
$ graphify query "which aws_instance resources have instance_type set to t3.micro"
```

Indexes Terraform block attributes (`ami`, `instance_type`, `cidr_block`, tags, nested blocks) as queryable, typed (bool/number/list/map) properties on resource/data/module nodes in `graph.json`, with secret-named values (`password`, `*secret*`, `*token*`, `*_key`, connection strings) automatically redacted before reaching the graph or the model.

Extracts JavaScript inside inline `<script>` blocks of PHP files as JS (functions and calls) under the PHP file node, with source lines mapped back to real file positions, mirroring the existing Vue/Svelte embedded-script handling.

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-08-20 CODE ↗

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

Diagram Design gains ridgeline, bubble, waterfall, Marimekko, and Streamgraph chart types plus Excalidraw import and new semantic patterns.

└──▷ **TRY IT**

Run the headless Chromium render linter against all shipped diagram examples to catch SVG clipping or overflow that source-only checks miss.

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

Run the adversarial Sankey verifier test suite to confirm all geometry contracts fire correctly before shipping a Sankey diagram change.

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

Run the ridgeline verifier and its adversarial test suite to validate stacked distribution examples after editing ridgeline markup.

```
$ python3 scripts/verify-ridgeline.py && python3 scripts/test-verify-ridgeline.py
```

Adds `ridgeline` as a `Line` variant for stacked distribution comparisons, with light, dark, and full editorial examples, `verify-ridgeline.py`, and `test-verify-ridgeline.py` verification scripts.

Adds `bubble` as a `Scatter` variant for three-value comparisons.

Adds `marimekko` as a `Treemap` variant.

Adds `Streamgraph` as a `Line` variant.

Adds a waterfall chart as a new visual type.

+6 more

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

AutoGPT

Feedback ⓘ autogpt-platform-beta-v0.8.0 NOTES ↗ RANK

AutoGPT automates multi-step workflows with continuous AI agents that users can build and deploy.

> AutoGPT Platform v0.8.0 ships a security-hardening overhaul — per-install secret generation, an encryption-key rotation CLI, and removal of legacy JWT verification — alongside a broad expansion of capabilities including MCP authentication support, a new capability registry, AutoPilot Voice Mode, E2B desktop sandboxes, ten new AI model integrations, skill package marketplace distribution, and a large build-out of the expert system (budgets, onboarding, standing work, shareable pages).

└──▷ **WHAT SHIPPED · 23 FEATURES** most completely described first ? what's the number?

01 Per-install secret generation with key rotation CLI

BREAKING

88

`ENCRYPTION_KEY`, `UNSUBSCRIBE_SECRET_KEY`, and `BETTER_AUTH_SECRET` no longer ship with values in `.env.default`; secrets are now generated per install via `make init-env`, and the backend refuses to start on the old published defaults. A new `cli rotate-encryption-key` subcommand (with `--apply` flag) re-encrypts stored data when rotating `ENCRYPTION_KEY`; installs running on the old defaults must run it before starting.

Rotate the encryption key on an existing self-hosted install that was running on the old published default, without losing stored integrations.

```
$ docker compose run --rm --no-deps -e OLD_ENCRYPTION_KEY
rest_server cli rotate-encryption-key
docker compose run --rm --no-deps -e OLD_ENCRYPTION_KEY
rest_server cli rotate-encryption-key --apply
```

— Names exact env vars, command and flag for migration. autogpt-platform-beta-v0.8.0

02 Capability registry and copilot discovery NEW

67

Adds a capability registry core and index, with copilot discovery switched to `find` / `describe` / `run_capability`.

— Names the three discovery primitives, no usage example. autogpt-platform-beta-v0.8.0

03 Interactive E2B desktop sandboxes NEW

62

Adds interactive E2B desktop sandboxes with live embedded streams, running copilot sandboxes on a dedicated 1 vCPU / 2 GiB desktop image.

— Names sandbox spec but no invocation path. autogpt-platform-beta-v0.8.0

04 **Skill packages and marketplace distribution** NEW 62

Adds skill packages: upload and download a skill as a zip, store/copy/delete a skill as its whole package, and publish and install skill packages to/from the marketplace. Also lists knowledge skills in the marketplace and enables installing them to AutoPilot.

— Names each package operation, no exact command shown. autogpt-platform-beta-v0.8.0

THINNER COVERAGE BELOW

05 **MCP authentication and protocol support** NEW 57

MCP servers now support Basic authentication, the platform speaks MCP protocol version `2026-07-28` alongside legacy MCP servers, and adds supported official MCP integrations.

— Names protocol version but not which integrations were added. autogpt-platform-beta-v0.8.0

06 **Ten new AI model integrations** NEW 53

Adds support for GPT-6 Astra, Opus 5, Claude Fable 5.1, Google Gemini 3.8 Flash, Alibaba Qwen 3.8 Max (0902), Meta Muse Spark 1.3, DeepSeek V4.1 Flash, Sakana Fugu Ultra v2, Inception Mercury 2.5, and advanced Anthropic model support.

— Names every model added, no usage detail given. autogpt-platform-beta-v0.8.0

07 **Expert budget tracking and spend thresholds** NEW 52

Adds an expert budget endpoint and run source, plus an expert spend threshold that pauses expert work for approval once spend reaches the configured limit.

— Describes threshold mechanism, endpoint not exactly named. autogpt-platform-beta-v0.8.0

08 **Discord bot mentions and impersonation alerts** NEW 48

Adds Discord bot support for mentioning server members and roles (never `@everyone`), and alerts and gates admin impersonation start via Discord.

— Names the exclusion rule, no setup steps given. autogpt-platform-beta-v0.8.0

09 **Chat-platform bot message controls** NEW 48

Lets AutoPilot edit its own chat-platform messages and adds a `NO_REPLY` signal for the chat-platform bot to stay silent instead of responding.

— Names the `NO_REPLY` signal, no trigger conditions given. autogpt-platform-beta-v0.8.0

10 **Expert-first onboarding wizard** NEW 45

Adds an expert-first onboarding wizard with brain-dump expert recommendations and generated expert avatars.

— Names wizard mechanism, no UI path given. autogpt-platform-beta-v0.8.0

11 **Legacy JWT verification removed** BREAKING 43

Removes the legacy HS256 JWT verification path; `JWT_VERIFY_KEY` in `backend/.env` is no longer read.

— Names the removed key but gives no migration guidance. autogpt-platform-beta-v0.8.0

12 **Dedicated shareable expert pages** NEW 43

Adds dedicated, shareable expert pages with canonical categories and a category filter.

— Names categories and filter, no page path given. autogpt-platform-beta-v0.8.0

13 **New workflow block types** NEW 42

Adds moderation blocks, TypeSafe Jev judgment blocks, and RMFG manufacturing blocks.

— Names three block types, no behavior described. autogpt-platform-beta-v0.8.0

14 **Linux and macOS appliance installer** NEW 35

Adds an appliance installer for Linux and macOS.

— States supported platforms only, no install steps given. autogpt-platform-beta-v0.8.0

15 **Agent run activation events** NEW 33

Records how each agent run started and emits activation events.

— States what's recorded, no event schema given. autogpt-platform-beta-v0.8.0

16 **Standing work routines for experts and Otto** NEW 33

Adds standing work (routines) for experts and Otto, which can be repeating or one-time.

— Describes recurrence options, no config detail given. autogpt-platform-beta-v0.8.0

17 **Microsoft 365 Copilot transport** NEW 28

Adds Microsoft 365 Copilot transport.

— Names the transport, no configuration detail given. autogpt-platform-beta-v0.8.0

18 **Open Graph previews for marketplace pages** NEW 27

Adds Open Graph link previews for marketplace pages.

— Names surface, no other detail given. autogpt-platform-beta-v0.8.0

19 **TrialNotificationDelivery table dropped** BREAKING 25

A database migration drops the `TrialNotificationDelivery` table.

— Bare mention of a dropped table, no context given. autogpt-platform-beta-v0.8.0

20 **AutoPilot Voice Mode** NEW 23

Adds AutoPilot Voice Mode.

— Bare feature name, no mechanism described. autogpt-platform-beta-v0.8.0

21 **Per-user A/B experiment arm tracking** NEW 23

Records A/B experiment arms per user.

— Bare statement, no experiments named. autogpt-platform-beta-v0.8.0

22 **Nine business core experts** NEW 23

Adds nine business core experts.

— Bare count, no names or capabilities given. autogpt-platform-beta-v0.8.0

23 **Automation credit history** NEW 21

Adds actionable automation credit history.

— Bare mention, no detail on what data is shown. autogpt-platform-beta-v0.8.0

└──▷ **BREAKING ON UPGRADE**

! `ENCRYPTION_KEY`, `UNSUBSCRIBE_SECRET_KEY`, and `BETTER_AUTH_SECRET` no longer ship with values in `.env.default`; the backend refuses to start on the old published default value. Installs that ran on the defaults must rotate their key using `cli rotate-encryption-key` / `cli rotate-encryption-key --apply` before starting.

! `JWT_VERIFY_KEY` in `backend/.env` is no longer read; the legacy HS256 JWT verification path has been removed.

! The `TrialNotificationDelivery` table is dropped by migration.

autogpt-platform-beta-v0.8.0 NOTES ↗ LATEST STABLE

AutoGPT Platform v0.8.0 adds voice mode, MCP Basic auth, skill packages, 8 new AI models, and per-install secret generation.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch autogpt-platform-beta-v0.8.0
https://github.com/Significant-Gravitas/AutoGPT.git
# already have the repo? check out this version:
$ git checkout autogpt-platform-beta-v0.8.0
```

└──▷ **TRY IT**

Rotate the encryption key on an existing self-hosted install that was running on the old published default, without losing stored integrations.

```
$ docker compose run --rm --no-deps -e OLD_ENCRYPTION_KEY
rest_server cli rotate-encryption-key
docker compose run --rm --no-deps -e OLD_ENCRYPTION_KEY rest_server
cli rotate-encryption-key --apply
```

Removes default values for `ENCRYPTION_KEY`, `UNSUBSCRIBE_SECRET_KEY`, and `BETTER_AUTH_SECRET` from `.env.default`; secrets are now generated per install via `make init-env`, and the backend refuses to start on the old published defaults.

Adds `cli rotate-encryption-key` subcommand (with `--apply` flag) to re-encrypt stored data when rotating `ENCRYPTION_KEY`.

Removes the legacy HS256 JWT verification path; `JWT_VERIFY_KEY` in `backend/.env` is no longer read.

Adds `NO_REPLY` signal for the chat-platform bot to stay silent instead of responding.

Supports Basic authentication for MCP servers.

+28 more

└──▷ **BREAKING ON UPGRADE**

- ! `ENCRYPTION_KEY`, `UNSUBSCRIBE_SECRET_KEY`, and `BETTER_AUTH_SECRET` no longer ship with values in `.env.default`; the backend refuses to start on the old published default value. Installs that ran on the defaults must rotate their key using `cli rotate-encryption-key` / `cli rotate-encryption-key --apply` before starting.
- ! `JWT_VERIFY_KEY` in `backend/.env` is no longer read; the legacy HS256 JWT verification path has been removed.
- ! The `TrialNotificationDelivery` table is dropped by migration.

WAS THIS USEFUL? 👍 👎

PydanticAI Feedback ⓘ v2.46.0 NOTES ↗ RANK

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

> PydanticAI's headline addition this window is the Choices()/Choice API for building structured outputs from option-sets known only at run time, alongside enum member docstrings surfaced in schemas, a new supports_text_output model-profile flag, realtime session playback control, Temporal event streaming, and a boolean-confidence threshold.

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

01 Choices() and Choice for run-time option sets NEW

95

Choices() accepts a mapping of keys to descriptions (or `Choice` objects) and returns a validated, typed output from an option-set known only at run time — usable as `output_type`, a Pydantic model field, or a tool parameter. `Choice` pairs a description with an optional `value=` so the model picks a key while the agent returns the associated value (e.g. a `Doc` instance) or calls the associated callable, eliminating a manual dispatch table.

Pick from a set of documents retrieved at run time — the model selects a key and the agent returns the matching `Doc` object.

python

```python
from pydantic_ai import Agent, Choice, Choices

class Doc:
    id: str
    title: str

docs = [
    Doc(id='rfc-6265', title='HTTP State Management Mechanism'),
    Doc(id='rfc-9110', title='HTTP Semantics'),
]

agent = Agent(
    'openai:gpt-4o',
    output_type=Choices({doc.id: Choice(doc.title, value=doc) for
doc in docs}),
)
result = agent.run_sync('Which RFC covers cookies?')
# result.output -> Doc(id='rfc-6265', ...)
```

Route an agent to UI actions known only at run time — the model picks an action key and the agent calls the bound callable directly.

python

```python
from functools import partial
from pydantic_ai import Agent, Choice, Choices

class Screen:
    def click(self, target: str) -> str:
        return f'clicked {target}'

screen = Screen()
targets = {'login': 'The Login button, top right.', 'forgot':
'The "forgot password" link.'}
actions = {
    t: Choice(desc, value=partial(screen.click, t))
    for t, desc in targets.items()
}

agent = Agent(
    'openai:gpt-4o',
    output_type=Choices(actions, description='Which action to
take on the screen.'),
)
result = agent.run_sync('Sign in as the admin.')
# result.output -> 'clicked login'
```

— Mechanism explained with two runnable code examples v2.46.0

02 **UseEnumMemberDocstrings mixin for enum schemas** NEW

85

`UseEnumMemberDocstrings` mixin forwards per-member docstrings on an `Enum` into tool and output schemas wherever the enum appears — in an output model, a tool parameter, or as a bare `Enum` — so the model sees a human-readable description for each option.

Describe each enum option to the model by adding `UseEnumMemberDocstrings` so per-member docstrings appear in the schema.

```python
from enum import Enum
from pydantic_ai import Agent
from pydantic_ai import UseEnumMemberDocstrings

class Resolution(UseEnumMemberDocstrings, Enum):
    refund = 'refund'
    """Return the customer's money."""
    replace = 'replace'
    """Send a working replacement unit."""

agent = Agent('openai:gpt-4o', output_type=Resolution)
result = agent.run_sync('The blender arrived broken. What should
we do?')
# result.output -> Resolution.replace
```

Give an enum's values human-readable descriptions sourced from docstrings so the LLM chooses more accurately.

```python
from enum import Enum
from pydantic_ai import UseEnumMemberDocstrings

class Priority(UseEnumMemberDocstrings, Enum):
    low = 'low'
    """Routine work with no deadline pressure."""
    high = 'high'
    """Time-sensitive items that block other work."""
```

— Named mixin with two worked examples v2.46.0

03 **RealtimeSession.wait_for_playback()** NEW 80

RealtimeSession.wait_for_playback() blocks until audio playback finishes before closing a realtime session, useful for ensuring a voice session has fully finished speaking in scripts or CI before exit.

Ensure a realtime voice session fully finishes speaking before your script exits — useful in CI or scripted demos.

```python
await session.wait_for_playback()
```

— Exact method call shown with runnable example v2.46.0

04 **event_stream_topic for Temporal workflow streaming** NEW

80

Adds `event_stream_topic` to `TemporalDurability` to stream agent events via Temporal Workflow Streams, letting downstream consumers react to agent events in real time without polling.

Stream agent events out of a Temporal workflow so downstream consumers can react in real time without polling.

python

```python
from pydantic_ai.durable_exec.temporal import TemporalDurability

agent = Agent(
    'openai:gpt-5.6-sol',
    capabilities=[
        TemporalDurability(event_stream_topic='agent-events'),
    ],
)
```

— Named config field with a runnable example v2.46.0

05 **supports_text_output model profile flag** NEW

70

Adds a `supports_text_output` field to `ModelProfile`; PydanticAI enforces `supports_text_output=False` at request-construction time, raising `UserError` before any network call when a model that cannot produce text is given a text-bearing output type (default text output, a union including text, an output function, or `str`). The same field lets `LLMJudge` and `GEval` run on models that do not support text output.

— Names field and error but gives no usage example v2.46.0

THINNER COVERAGE BELOW

06 **typesafe_boolean_threshold parameter** NEW

40

Adds a `typesafe_boolean_threshold` parameter to control the confidence level at which a yes/no model response is treated as `True`.

— Named parameter but no default or example given v2.46.0

07 **TypeSafeModel union output selection** IMPROVED

30

`TypeSafeModel` can now fill a union of output types by first selecting the appropriate type.

— One-line description with no mechanism or example v2.46.0

v2.46.0 NOTES ↗ LATEST STABLE

PydanticAI v2.46.0 adds RealtimeSession.wait_for_playback(), Choices helper, typesafe_boolean_threshold, Temporal event streaming, and UseEnumMemberDocstrings support.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.46.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.46.0
```

└──▷ **USE IT**

Ensure a realtime voice session fully finishes speaking before your script exits — useful in CI or scripted demos.

```python
await session.wait_for_playback()
```

Stream agent events out of a Temporal workflow so downstream consumers can react in real time without polling.

```python
from pydantic_ai.durable_exec.temporal import TemporalDurability

agent = Agent(
    'openai:gpt-5.6-sol',
    capabilities=[
        TemporalDurability(event_stream_topic='agent-events'),
    ],
)
```

Give an enum's values human-readable descriptions sourced from docstrings so the LLM chooses more accurately.

```python
from enum import Enum
from pydantic_ai import UseEnumMemberDocstrings

class Priority(UseEnumMemberDocstrings, Enum):
    low = 'low'
    """Routine work with no deadline pressure."""
    high = 'high'
    """Time-sensitive items that block other work."""
```

Adds RealtimeSession.wait_for_playback() method to block until audio playback finishes before closing a realtime session.

Adds `typesafe_boolean_threshold` parameter to control the confidence level at which a yes/no model response is treated as `True`.

Adds `event_stream_topic` to `TemporalDurability` to stream agent events via Temporal Workflow Streams.

Adds `supports_text_output` field to `ModelProfile`, enabling `LLMJudge` and `GEval` to run on models that do not support text output.

Adds a `Choices` helper class for defining a set of described options assembled at run time.

+2 more

WAS THIS USEFUL?  

CopilotKit

Feedback ⓘ v1.73.0 NOTES ↗ RANK

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

> CopilotKit v1.73.0 adds automatic Learning skill delivery to BuiltInAgent and strengthens the runtime with more resilient SSE streaming and richer telemetry.

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

01 **Automatic Learning skill delivery to BuiltInAgent** NEW 33

BuiltInAgent now receives Learning skills automatically, enabling self-improving agent behavior without manual skill wiring.

— Names the mechanism but no config or API details. v1.73.0

02 **More resilient SSE streaming in runtime** IMPROVED 20

The runtime improves SSE streaming resilience for more reliable agent-to-UI event delivery.

— Describes benefit only, no mechanism or config named. v1.73.0

03 **Richer runtime telemetry** IMPROVED 18

The runtime adds richer telemetry for deeper observability into agent execution.

— Generic description, no specific metrics or fields named. v1.73.0

v1.73.0 NOTES ↗ LATEST STABLE

CopilotKit v1.73.0 adds automatic Learning skill delivery to BuiltInAgent and richer runtime telemetry with more resilient SSE streaming.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.73.0
https://github.com/CopilotKit/CopilotKit.git
# already have the repo? check out this version:
$ git checkout v1.73.0
```

Delivers Learning skills automatically to BuiltInAgent, enabling self-improving agent behavior without manual skill wiring.

> Improves SSE streaming resilience in the runtime for more reliable agent-to-UI event delivery.
>
> Adds richer runtime telemetry for deeper observability into agent execution.

WAS THIS USEFUL?  

CopilotKit OpenBot

Feedback ⓘ v0.0.13 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 v0.0.13 adds Anthropic and Gemini support to the LangGraph Bot, expands the Google Drive connector to shared drives with memory-capped downloads, fixes MCP embedded-resource text passthrough, and ships signed desktop builds for Windows and Mac.

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

01 **Google Drive connector shared-drive and memory improvements**

IMPROVED 80

Extends the Google Drive connector to reach files in shared drives by adding shared-drive support to `get_file_metadata`, `read_file_content`, `search_files`, and `list_recent_files` requests. Also excludes trashed files from `search_files` and `list_recent_files` results, and caps memory use in `read_file_content` by streaming and cancelling large file downloads after the 20,000-character limit is reached instead of loading the full file into memory.

— Names every affected method and the concrete memory limit v0.0.13

02 **Anthropic and Gemini support for LangGraph Bot** NEW

70

Supports `BOT_PROVIDER=anthropic` and `BOT_PROVIDER=google` for the LangGraph Bot by folding multiple system prompts into one, so runs on Anthropic and Gemini no longer fail before the model is asked.

— Names exact env values but no further mechanism detail v0.0.13

03 **MCP embedded-resource text passthrough** IMPROVED 68

Passes the text content of MCP embedded resources to the Bot as a text part, so tools like GitHub MCP server's `get_file_contents` now deliver file contents instead of `[resource]`.

— Clear before/after behaviour with named tool example v0.0.13

04 **BOT_MODEL whitespace trimming and fallback** IMPROVED

67

Trims whitespace from `BOT_MODEL` in the proof-of-concept Bot, falling back to the default when the value is empty or padded, preventing silent failures on direct environment variable injection (e.g. Kubernetes manifests or `docker run -e`).

Set a model name directly in a Kubernetes manifest — v0.0.13 trims surrounding whitespace and falls back to the default instead of silently starting with a broken name.

```yaml
- name: BOT_MODEL
  value: " gpt-4o "
```

— Includes runnable config example showing the fix v0.0.13

THINNER COVERAGE BELOW

05 **Sidebar toggle for non-Latin keyboards** IMPROVED 53

Adds Ctrl+B / ⌘B sidebar toggle support for non-Latin keyboard layouts (Russian, Greek) by reading the physical key when the layout writes a character outside ASCII.

— Names shortcut and affected layouts but narrow scope v0.0.13

06 **Signed desktop builds for Windows and Mac** NEW 52

Delivers desktop builds for Windows x64 (signed installer) and Mac Apple Silicon (ad-hoc signed DMG) via GitHub Actions from release commit `35ad699f6f760d90233178b63517241573520cbe`.

— Names platforms and commit but no usage path v0.0.13

v0.0.13 NOTES ↗ LATEST STABLE

OpenBot v0.0.13 expands Google Drive shared-drive access, fixes MCP embedded-resource text passthrough, and adds Anthropic/Gemini LangGraph Bot support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.0.13
https://github.com/CopilotKit/OpenBot.git
# already have the repo? check out this version:
$ git checkout v0.0.13
```

└──▷ USE IT

Set a model name directly in a Kubernetes manifest — v0.0.13 trims surrounding whitespace and falls back to the default instead of silently starting with a broken name.

```yaml
- name: BOT_MODEL
  value: " gpt-4o "
```

> Supports `BOT_PROVIDER=anthropic` and `BOT_PROVIDER=google` for the LangGraph Bot by folding multiple system prompts into one, so runs on Anthropic and Gemini no longer fail before the model is asked.
>
> Trims whitespace from `BOT_MODEL` in the proof-of-concept Bot, falling back to the default when the value is empty or padded, preventing silent failures on direct environment variable injection (e.g. Kubernetes manifests or `docker run -e`).
>
> Passes the text content of MCP embedded resources to the Bot as a text part, so tools like GitHub MCP server's `get_file_contents` now deliver file contents instead of `[resource]`.
>
> Extends the Google Drive connector to reach files in shared drives by adding shared-drive support to `get_file_metadata`, `read_file_content`, `search_files`, and `list_recent_files` requests.
>
> Excludes trashed files from `search_files` and `list_recent_files` results in the Google Drive connector.
>
> +3 more

WAS THIS USEFUL? 👍 👎

Microsoft Agent Framework

Feedback ⓘ dotnet-1.22.0

NOTES ↗ RANK

The Microsoft Agent Framework builds, orchestrates, and deploys agents and multi-agent workflows in Python and .NET, consolidating AutoGen and Semantic Kernel.

> dotnet-1.22.0 adds an AsIChatClient interop extension and per-tool AgentModeProvider policies, while introducing several breaking changes to session, MCP, and approval-binding behavior alongside smaller refinements to telemetry, Purview coverage, and skill loading.

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

01 Public OpenTelemetryAgent.DefaultSourceName property NEW

75

Exposes `OpenTelemetryAgent.DefaultSourceName` as a public static property so custom telemetry configurations can subscribe to the framework's default `ActivitySource` by name.

Reference `OpenTelemetryAgent.DefaultSourceName` when wiring up a custom ActivitySource listener so your telemetry pipeline captures all agent traces by name.

csharp

```
using Microsoft.Agents.AI;
using System.Diagnostics;

// Subscribe to the framework's default telemetry source
var listener = new ActivityListener
{
    ShouldListenTo = source => source.Name ==
OpenTelemetryAgent.DefaultSourceName,
    Sample = (ref ActivityCreationOptions<ActivityContext> _) =>
ActivitySamplingResult.AllData,
    ActivityStarted = activity => Console.WriteLine($"Started:
{activity.DisplayName}"),
    ActivityStopped = activity => Console.WriteLine($"Stopped:
{activity.DisplayName}")
};
ActivitySource.AddActivityListener(listener);
```

— Includes a runnable code example wiring the listener dotnet-1.22.0

02 AgentSessionStore promoted to public abstractions BREAKING

60

Promotes `AgentSessionStore` into `Agents.AI.Abstractions`, making it part of the public abstractions layer; code referencing it from its previous namespace will require an update.

— Names the target namespace and migration impact `dotnet-1.22.0`

THINNER COVERAGE BELOW

03 **AIAgent to IChatClient interop via AsIChatClient** NEW

55

Adds an `AsIChatClient` extension method that exposes an `AIAgent` as an `IChatClient`, enabling interop with the `Microsoft.Extensions.AI` ecosystem.

— Names the method but no usage example given `dotnet-1.22.0`

04 **Approval Binding for always-approval responses** BREAKING

50

Binds `always approval` responses to surfaced approval requests, improving replay support with Approval Binding. This introduces breaking changes to the binding contract, so existing approval-binding integrations must be updated.

— Describes mechanism and breaking impact but no migration steps `dotnet-1.22.0`

05 **Per-run tool scoping for ChatClientAgent** IMPROVED 50

Passes `ChatClientAgent` tools per run only rather than setting them on the underlying FICC, avoiding shared-state side effects.

— Names the class and mechanism but no code sample `dotnet-1.22.0`

06 **Sticky Foundry delegated identity on sessions** BREAKING

50

Foundry delegated user identity is now sticky on `AgentSession`; code that previously set or changed the delegated identity between invocations on the same session will behave differently.

— Names the affected type and behavior change `dotnet-1.22.0`

07 **Request-scoped Python Foundry agent factories** BREAKING

45

Python: request-scoped Foundry agent factories replace the previous factory model; existing Python Foundry agent factory usage requires migration.

— Notes migration need but no code example `dotnet-1.22.0`

08 **Token counts preserved in Foundry Hosting** IMPROVED 35

Preserves cached and reasoning token counts in Foundry Hosting, surfacing those values in run results.

— Names Foundry Hosting and token fields but no exact field names dotnet-1.22.0

09 **Refined skill frontmatter parsing** IMPROVED 30

Refines skill frontmatter parsing in .NET and Python for more robust declarative agent skill loading.

— Improvement noted without specific parsing rules dotnet-1.22.0

10 **Middleware preserved for dynamic tools** IMPROVED 25

Preserves harness middleware for dynamically registered tools, ensuring middleware pipelines apply consistently.

— Behavioral fix described only in prose dotnet-1.22.0

11 **Expanded Purview content types** IMPROVED 20

Adds more content types to Purview handling, broadening data-governance coverage across agent messages.

— No specific content types or mechanism named dotnet-1.22.0

BREAKING ON UPGRADE

- ! `AgentSessionStore` is promoted into `Agents.AI.Abstractions`; any code referencing it from its previous namespace will require an update.
- ! Replay support with Approval Binding has breaking changes — existing approval-binding integrations must be updated to the new binding contract.
- ! Provider-backed MCP sessions are now scoped per invocation rather than shared; code that relied on session state persisting across invocations will break.
- ! Foundry delegated user identity is now sticky on `AgentSession`; code that previously set or changed the delegated identity between invocations on the same session will behave differently.
- ! Python: request-scoped Foundry agent factories replace the previous factory model; existing Python Foundry agent factory usage requires migration.

dotnet-1.22.0 NOTES ↗ LATEST STABLE

dotnet-1.22.0 adds AsIChatClient extension, per-tool AgentModeProvider controls, expanded Purview content types, and several breaking session/MCP changes.

GET THIS VERSION ▾

```
$ git clone --branch dotnet-1.22.0
https://github.com/microsoft/agent-framework.git
# already have the repo? check out this version:
$ git checkout dotnet-1.22.0
```

└──▷ **USE IT**

Reference `OpenTelemetryAgent.DefaultSourceName` when wiring up a custom ActivitySource listener so your telemetry pipeline captures all agent traces by name.

```csharp
using Microsoft.Agents.AI;
using System.Diagnostics;

// Subscribe to the framework's default telemetry source
var listener = new ActivityListener
{
    ShouldListenTo = source => source.Name ==
OpenTelemetryAgent.DefaultSourceName,
    Sample = (ref ActivityCreationOptions<ActivityContext> _) =>
ActivitySamplingResult.AllData,
    ActivityStarted = activity => Console.WriteLine($"Started:
{activity.DisplayName}"),
    ActivityStopped = activity => Console.WriteLine($"Stopped:
{activity.DisplayName}")
};
ActivitySource.AddActivityListener(listener);
```

Adds `AsIChatClient` extension method to expose an `AIAgent` as an `IChatClient`, enabling interop with the `Microsoft.Extensions.AI` ecosystem.

Exposes `OpenTelemetryAgent.DefaultSourceName` as a public static property for use in custom telemetry configuration.

Adds per-tool `AgentModeProvider` controls, allowing different agent mode policies to be set on individual tools rather than globally.

Promotes `AgentSessionStore` into `Agents.AI.Abstractions`, making it part of the public abstractions layer.

Adds more content types to Purview handling, broadening data-governance coverage across agent messages.

+5 more

└──▷ **BREAKING ON UPGRADE**

- ! `AgentSessionStore` is promoted into `Agents.AI.Abstractions`; any code referencing it from its previous namespace will require an update.
- ! Replay support with Approval Binding has breaking changes — existing approval-binding integrations must be updated to the new binding contract.
- ! Provider-backed MCP sessions are now scoped per invocation rather than shared; code that relied on session state persisting across invocations will break.
- ! Foundry delegated user identity is now sticky on `AgentSession`; code that previously set or changed the delegated identity between invocations on the same session will behave differently.
- ! Python: request-scoped Foundry agent factories replace the previous factory model; existing Python Foundry agent factory usage requires migration.

WAS THIS USEFUL? 👍 👎

Cloudflare Agents

Feedback ⓘ agents@0.24.0 NOTES ↗ RANK

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

> Cloudflare Agents 0.24.0 adds a durable Queue Lifecycle capability, a Cap'n Web RPC transport, and a standalone State capability that any Lifecycle host can use without inheriting Agent, alongside breaking changes across Lifecycle internals, WebSockets, and workflow-notification retry behavior.

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

01 Queue Lifecycle capability for background jobs

BREAKING

90

Adds the Queue Lifecycle capability (`agents/queue`) for durable background work: each push() call enqueues a job with an optional stable `id` for upsert and per-item `retry` . `dequeue` , `dequeueAll` , `dequeueAllByCallback` , `getQueue` , and `getQueues` are now asynchronous — callers that treated them as synchronous will break — and `QueueItem.createdAt` replaces `created_at` . On upgrade, the `cf_agents_queues` table is dropped on next start with rows migrated into the new job queue; deployments must pass through 0.24.0 before a future minor release removes the migration path.

— Names every method, field rename, and migration table agents@0.24.0

02 WebSockets capability supports plain Durable Objects

BREAKING

90

The `WebSockets` capability gains a `callables` option so a plain Durable Object (not inheriting Agent) works with `useAgent` and `AgentClient` : `protocol: false` hands the connect sequence to the host, and `callables` exposes an `RpcTarget` whose prototype methods become the host's remote interface. It also gains a `state` option accepting a State capability instance, enabling `useAgent().state` and setState() against a plain Durable Object, with per-connection `readonly` and `protocol` flags decided at accept time. New methods sendIdentity(), sendState(), applyStateFrame(), isReadonly()/setReadonly(), isProtocolEnabled()/setProtocolEnabled(), and `registerInternalConnectionKeys` support hosts that drive the connect sequence themselves. The `identity` option introduced in 0.23.0 is replaced by `protocol` .

— Enumerates every new option and method by name agents@0.24.0

03 Cap'n Web transport for useAgent and AgentClient

BREAKING

86

Adds Cap'n Web transport to useAgent({ transport }) and AgentClient({ transport }): passing `'capnweb'` runs a session that carries protocol frames natively — call()/ `stub` invoke callables directly, an `RpcTarget` result comes back as a live stub, a `ReadableStream` streams, and calls pipeline. This replaces the experimental `?__agents_rpc=capnweb` endpoint from 0.23.0, which is removed.

Connect a React client to an agent over Cap'n Web for native RPC — live stubs, streaming results, and call pipelining.

typescript

```
import { useAgent } from 'agents/react';

function App() {
  const agent = useAgent({
    agent: 'CounterAgent',
    transport: 'capnweb',
  });

  return <button onClick={() => agent.stub.increment()}>+
</button>;
}
```

— Runnable example plus removal of prior endpoint noted `agents@0.24.0`

04 **Standalone State Lifecycle capability** NEW

83

Adds the opt-in State Lifecycle capability so any Lifecycle host gets durable, validated state without inheriting Agent: it accepts `initialState`, `validateStateChange`, and `onChanged` options and owns the `cf_agents_state` storage row.

Give any Lifecycle host durable, validated state without inheriting Agent.

typescript

```
import { State } from 'agents';

const state = new State({
  initialState: { count: 0 },
  validateStateChange: (next, source) => {
    if (next.count < 0) throw new Error('count cannot be
negative');
    return next;
  },
  onChanged: (state, source) => console.log('state changed',
state),
});
```

— Named options plus a runnable instantiation example `agents@0.24.0`

05 Lifecycle capability internals reworked BREAKING

65

`LifecycleServices` now exposes `name` and `className` on the service object. Lifecycle capabilities declare traffic ownership with `claims: 'selective' | 'catch-all'` instead of passing `{ fallback: true }` to lifecycle.use(), and `LifecycleUseOptions` is removed. LifecycleServices.starting() is replaced by status(), which returns `'zero' | 'starting' | 'started'`; code still calling .starting() will throw.

— Names replaced fields but gives no migration example `agents@0.24.0`

THINNER COVERAGE BELOW

06 Workflow-notification retry and timeout changes IMPROVED

55

Workflow-notification delivery now gives up after twelve hours of continuous failure (previously never gave up) and retries with backoff capped at ten minutes (previously five), reporting failure through `onError`. The `cf_think_workflow_notifications` table is dropped on next start with rows migrated, under the same upgrade-path constraint as the queue table.

— Concrete before/after numbers but no direct API to act on `agents@0.24.0`

07 ResumableStream cutover detection NEW

25

Exposes `ResumableStream.pendingCutoverId` so callers can detect whether a finished stream still awaits its cutover.

— Bare field name with no usage context `agents@0.24.0`

└──▷ **BREAKING ON UPGRADE**

! The experimental `?__agents_rpc=capnweb` endpoint from 0.23.0 is removed; use `transport: 'capnweb'` in useAgent() or AgentClient() instead.

! `dequeue`, `dequeueAll`, `dequeueAllByCallback`, `getQueue`, and `getQueues` are now asynchronous — callers that treated them as synchronous will break.

! `QueueItem.created_at` is renamed to `createdAt` — code reading the old field will silently get `undefined`.

! LifecycleServices.starting() is replaced by status() which returns `'zero' | 'starting' | 'started'` — any code calling .starting() will throw.

! `LifecycleUseOptions` is removed and the `{ fallback: true }` argument to lifecycle.use() is no longer accepted; capabilities must instead declare `claims: 'selective' | 'catch-all'`.

! The `identity` option on `WebSockets` (introduced in 0.23.0) is replaced by `protocol`.

! The `cf_agents_queues` table is dropped on next start (rows are migrated into the job queue); deployments skipping 0.24.0 must upgrade through it before the next minor release

removes the migration path.

! The `cf_think_workflow_notifications` table is dropped on next start (rows migrated); same upgrade-path constraint applies.

! Workflow-notification delivery now gives up after twelve hours of continuous failure (previously never gave up) and retries with backoff capped at ten minutes (previously five), reporting failure through `onError`.

agents@0.24.0 NOTES ↗ LATEST STABLE

Cloudflare Agents 0.24.0 adds a Queue Lifecycle capability, Cap'n Web transport, and a standalone State capability.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch agents@0.24.0
https://github.com/cloudflare/agents.git
# already have the repo? check out this version:
$ git checkout agents@0.24.0
```

└──▷ **USE IT**

Connect a React client to an agent over Cap'n Web for native RPC — live stubs, streaming results, and call pipelining.

typescript

```typescript
import { useAgent } from 'agents/react';

function App() {
  const agent = useAgent({
    agent: 'CounterAgent',
    transport: 'capnweb',
  });

  return <button onClick={() => agent.stub.increment()}>+</button>;
}
```

Give any Lifecycle host durable, validated state without inheriting Agent.

```typescript
import { State } from 'agents';

const state = new State({
  initialState: { count: 0 },
  validateStateChange: (next, source) => {
    if (next.count < 0) throw new Error('count cannot be negative');
    return next;
  },
  onChanged: (state, source) => console.log('state changed', state),
});
```

Adds the `Queue` Lifecycle capability (`agents/queue`) for durable background work: each push() call enqueues a job with optional stable `id` (upsert) and per-item `retry`; `dequeue`, `dequeueAll`, `dequeueAllByCallback`, `getQueue`, and `getQueues` are now async; `QueueItem.createdAt` replaces `created_at`.

Adds Cap'n Web transport to useAgent({ transport }) and AgentClient({ transport }): pass `'capnweb'` to run a Cap'n Web session that carries protocol frames natively — call()/ `stub` invoke callables directly, an `RpcTarget` result is a live stub, a `ReadableStream` streams, and calls pipeline.

Adds `callables` option to `WebSockets` so a plain Durable Object (not inheriting `Agent`) works with `useAgent` and `AgentClient`: `protocol: false` hands the connect sequence to the host, and `callables` exposes an `RpcTarget` whose prototype methods are the host's remote interface.

Adds the opt-in `State` Lifecycle capability so any Lifecycle host gets durable, validated state without inheriting `Agent`: accepts `initialState`, `validateStateChange`, and `onChanged` options and owns the `cf_agents_state` storage row.

Adds `state` option to the `WebSockets` capability accepting a `State` capability instance, enabling `useAgent().state` and setState() against a plain Durable Object with per-connection `readonly` and `protocol` flags decided at accept time.

+4 more

└─▷ **BREAKING ON UPGRADE**

! The experimental `?__agents_rpc=capnweb` endpoint from 0.23.0 is removed; use `transport: 'capnweb'` in useAgent() or AgentClient() instead.

! `dequeue`, `dequeueAll`, `dequeueAllByCallback`, `getQueue`, and `getQueues` are now asynchronous — callers that treated them as synchronous will break.

- ! `QueueItem.created_at` is renamed to `createdAt` — code reading the old field will silently get `undefined`.
- ! LifecycleServices.starting() is replaced by status() which returns `'zero' | 'starting' | 'started'` — any code calling .starting() will throw.
- ! `LifecycleUseOptions` is removed and the `{ fallback: true }` argument to lifecycle.use() is no longer accepted; capabilities must instead declare `claims: 'selective' | 'catch-all'`.
- ! The `identity` option on `WebSockets` (introduced in 0.23.0) is replaced by `protocol`.
- ! The `cf_agents_queues` table is dropped on next start (rows are migrated into the job queue); deployments skipping 0.24.0 must upgrade through it before the next minor release removes the migration path.
- ! The `cf_think_workflow_notifications` table is dropped on next start (rows migrated); same upgrade-path constraint applies.
- ! Workflow-notification delivery now gives up after twelve hours of continuous failure (previously never gave up) and retries with backoff capped at ten minutes (previously five), reporting failure through `onError`.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

HeyGen HyperFrames

Feedback ⓘ v0.8.50 NOTES ↗ RANK

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

> HyperFrames shipped a cluster of Studio editor upgrades this window: timeline clip-group duplication, a magnetic main track, a live browsable catalog grid, more precise edit-decline reporting, and a new ask-agent entry point from picked elements.

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

01 **Main track gap-closing and clip start behavior** IMPROVED 60

Studio's main track is now magnetic, automatically closing gaps when clips are deleted, and the first clip placed on an empty main track now starts at position 0 instead of an arbitrary offset.

– Two concrete before/after track behaviors, no config surface named v0.8.48 v0.8.49

THINNER COVERAGE BELOW

02 **Timeline clip group copy/paste/duplicate** NEW 55

Studio's timeline now supports copy, paste, and duplicate operations for clip groups, letting operators duplicate multi-clip selections directly on the timeline.

– Names the three operations but no shortcut or UI path v0.8.50

03 **Studio timeline row structure and agent clip visibility** IMPROVED 50

The Studio timeline now shows one row per top-level element with no child rows displayed, simplifying the timeline view. Separately, the agent now sees every timeline clip, including audio clips, which were previously invisible to it.

– Explains what changed but not the underlying cause v0.8.50

04 **Edit decline reporting with specific causes** IMPROVED 50

The editor now reports the real reason an edit was declined, choosing from one of four specific causes instead of returning one generic error.

– Names four-cause mechanism but not the causes themselves v0.8.49

05 **Catalog live browsable grid** IMPROVED 45

The catalog is now a live grid that can be browsed interactively, replacing what was previously a static index.

— Describes the change but no navigation or filter details v0.8.49

06 **Ask-agent flow from picked elements** NEW 45

Host applications can now open the ask-agent flow directly from a picked element, rather than requiring a separate entry point.

— Clear trigger described but no API or flag named v0.8.48

07 **Auto-open root composition on load** IMPROVED 35

Projects now automatically open their root composition on load, rather than requiring a manual step.

— Simple behavior change with no further mechanism given v0.8.48

08 **Segment ease control visible at rest** IMPROVED 35

The segment ease control is now visible and discoverable at rest, without requiring user interaction to reveal it.

— States visibility change only, no interaction path given v0.8.48

09 **Per-project UI preference scoping** NEW 30

UI preferences in Studio are now scoped per project rather than shared globally.

— Bare description, no mechanism or scope detail v0.8.50

v0.8.50 NOTES ↗ LATEST STABLE

HyperFrames v0.8.50 adds timeline copy/paste/duplicate for clip groups, preview ruler and safe-margin toggles, and per-project UI preference scoping.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.50 https://github.com/heygen-
com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.50
```

Adds copy, paste, and duplicate operations for clip groups on the Studio timeline.

Adds per-project UI preference scoping in Studio.

The Studio timeline now shows one row per top-level element with no child rows.

The agent now sees every timeline clip, including audio clips.

v0.8.49 NOTES ↗ STABLE

HyperFrames v0.8.49 makes the catalog a live browsable grid and sharpens Studio edit-decline reporting.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.49 https://github.com/heygen-
com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.49
```

The catalog is now a live grid you browse, not a static index.

The editor now reports the real reason an edit was declined, one of four causes, instead of one generic one.

The first clip on an empty main track now starts at 0.

v0.8.48 NOTES ↗ STABLE

HyperFrames v0.8.48 adds a magnetic main track, ask-agent flow from picked elements, and auto-opens the root composition on load.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.48 https://github.com/heygen-
com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.48
```

Studio operators get a magnetic main track that automatically closes gaps when clips are deleted.

Host apps can open the ask-agent flow directly from a picked element.

Projects now open their root composition automatically on load.

The segment ease control is visible and discoverable at rest without requiring user interaction.

WAS THIS USEFUL? 👍 👎

NVIDIA Megatron Bridge

Feedback ⓘ v0.6.2 NOTES ↗

RANK

Megatron Bridge connects Hugging Face model workflows with Megatron Core for distributed model training and checkpoint conversion.

> Megatron Bridge v0.6.2 adds an experimental NCCL EP backend for Hopper/Blackwell and automatic GCP NCCL plugin detection on select machine types, alongside a wave of new and improved performance recipes for Nemotron, DeepSeek, and Qwen3 models.

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

01 **New performance recipes for Nemotron, DeepSeek-V4-Flash, and Qwen3-235B**

NEW 85

Adds Nemotron 3.5 Lightning BF16, MXFP8, and NVFP4 performance recipes for B200, B300, and GB300, including a GB300 MXFP8 FSDP variant. Adds Nemotron 3 Ultra NVFP4 performance recipes for 256-GPU GB200/GB300 systems with FSDP/HSDP and activation offloading. Adds DeepSeek-V4-Flash MXFP8 performance recipes for B200/B300 and GB200/GB300 with expert-parallelism and batch-size tuning. Adds Qwen3-235B NVFP4 recipes for B200/B300/GB200/GB300 with CuTeDSL fused grouped MLP, HybridEP communication overlap, and full-iteration CUDA graphs.

— Names precisions, hardware and mechanisms for each recipe. v0.6.2

02 **GCP NCCL plugin auto-detection** NEW 75

Installs the GCP NCCL plugin package in the container; on A3 Ultra, A4, A4X, and A4X Max machine types the container automatically detects and activates GCP NCCL network, tuner, and environment plugins for optimized multi-node communication.

— Names exact machine types and plugin behavior, no command shown. v0.6.2

03 **Improved DeepSeek-V3 and Nemotron GB200/GB300 recipes**

IMPROVED 75

Improves the DeepSeek-V3 GB300 NVFP4 recipe with CuTeDSL fused grouped MLP, MLA fusion, FP8 attention, and full-iteration CUDA graphs. Improves Nemotron 3 Nano and Super GB200/GB300 MXFP8 recipes with CuTeDSL fused grouped MLP and FP8 attention.

— Lists concrete optimization techniques applied per recipe. v0.6.2

THINNER COVERAGE BELOW

04 **Experimental NCCL EP backend for Hopper/Blackwell** NEW

50

Adds experimental NCCL EP (endpoint) backend support for Hopper and Blackwell GPUs, with GPT-OSS and Nemotron benchmark examples.

— States scope and examples but no setup steps. v0.6.2

v0.6.2 NOTES ↗ LATEST STABLE

Megatron Bridge v0.6.2 adds GCP NCCL auto-detection and experimental NCCL EP backend for Hopper/Blackwell.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.6.2 https://github.com/NVIDIA-NeMo/Megatron-Bridge.git
# already have the repo? check out this version:
$ git checkout v0.6.2
```

Adds experimental NCCL EP (endpoint) backend support for Hopper and Blackwell GPUs, with GPT-OSS and Nemotron benchmark examples.

Installs the GCP NCCL plugin package in the container; on A3 Ultra, A4, A4X, and A4X Max machine types the container automatically detects and activates GCP NCCL network, tuner, and environment plugins for optimized multi-node communication.

Adds Nemotron 3.5 Lightning BF16, MXFP8, and NVFP4 performance recipes for B200, B300, and GB300, including a GB300 MXFP8 FSDP variant.

Adds Nemotron 3 Ultra NVFP4 performance recipes for 256-GPU GB200/GB300 systems with FSDP/HSDP and activation offloading.

Adds DeepSeek-V4-Flash MXFP8 performance recipes for B200/B300 and GB200/GB300 with expert-parallelism and batch-size tuning.

+3 more

WAS THIS USEFUL? 👍 👎

NVIDIA Megatron-LM

Feedback ⓘ core_v0.19.2 NOTES ↗ RANK

Megatron-LM trains transformer models at scale with tensor, pipeline, sequence, and context parallelism, and ships Megatron Core as a reusable library.

> Megatron-LM core_v0.19.2 adds FP4 precision support to pipeline-parallel overlap and bumps its TransformerEngine dependency.

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

01 **FP4 support in 1F1B all-to-all overlap** NEW 35

Adds FP4 precision support in the 1F1B (one-forward-one-backward) all-to-all overlap used for pipeline-parallel training.

— Names mechanism but no config flag or usage detail core_v0.19.2

02 **TransformerEngine dependency upgrade** IMPROVED 30

Upgrades the TransformerEngine dependency from 2.17 to 2.18.

— Version bump named but no changelog detail core_v0.19.2

core_v0.19.2 NOTES ↗ LATEST STABLE

Megatron-LM core_v0.19.2 adds FP4 support in 1F1B A2A overlap and upgrades TransformerEngine to 2.18.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch core_v0.19.2
https://github.com/NVIDIA/Megatron-LM.git
# already have the repo? check out this version:
$ git checkout core_v0.19.2
```

Adds FP4 precision support in 1F1B (one-forward-one-backward) all-to-all overlap for pipeline-parallel training.

Upgrades TransformerEngine dependency from 2.17 to 2.18.

WAS THIS USEFUL? 👍 👎

SkyPilot

Feedback ⓘ v0.13.2rc1 NOTES ↗ RANK

SkyPilot runs AI training and inference jobs across clouds and Kubernetes clusters, picking cheaper or available capacity and handling spot preemption.

> SkyPilot's v0.13.2rc1 release focuses on multi-tenant API server isolation, deeper Slurm cluster visibility and job execution controls, a reworked dashboard Infra page, and smaller fixes across Kueue provisioning, Kubernetes diagnostics and RunPod credentials.

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

01 **Slurm job execution and visibility improvements** IMPROVED 78

Adds SlurmClient.get_all_jobs_info() for per-node job attribution across Slurm clusters, enriches Slurm node info with CPU/memory allocation and sampled usage metrics, submits Slurm jobs as authenticated users, and unsets step-scoped `SLURM_*` environment variables before running user scripts.

— Names method and env vars, groups related backend changes v0.13.2rc1

02 **Kueue-aware provisioning timeout** IMPROVED 65

Excludes Kueue admission wait time from `provision_timeout` so queue-gated pods are not incorrectly timed out.

— Names config key and explains exact behavior fixed v0.13.2rc1

03 **Dashboard Infra page rework** IMPROVED 60

The dashboard Infra page is reworked with a unified table and per-GPU-type summary, breaks Slurm clusters down by partition, shows CPU-only Slurm clusters, and keeps unreachable Slurm clusters listed rather than hiding them.

— Clear UI destination named but no exact navigation steps v0.13.2rc1

THINNER COVERAGE BELOW

04 **Launch requests queue on lock contention** IMPROVED 55

Launch requests now park as `WAITING` on cluster lock contention instead of blocking a worker thread.

— Names state and behavior change, no user-facing action v0.13.2rc1

05 **GCP federated credentials for GCS copy** NEW 45

Supports GCP federated credentials for GCS `COPY` operations.

— Names the operation but no config detail v0.13.2rc1

06 **Dashboard table filtering and pagination** IMPROVED 45

Adds filtering to the dashboard volumes page and persists the rows-per-page preference for the cluster jobs and pools tables.

— Names affected pages but lacks mechanism detail v0.13.2rc1

07 **Richer RECOVERING job events** IMPROVED 45

Includes exit code and log pointer in RECOVERING job events for user-job failure restarts.

— Names event fields but no access path given v0.13.2rc1

08 **Kubernetes cluster diagnostics** IMPROVED 45

Surfaces which node a terminated Kubernetes cluster lost, and displays actual pod resource requests when they diverge from the resource spec.

— Describes two diagnostic improvements without exact UI/API path v0.13.2rc1

09 **Multiple isolated local API servers** NEW 30

SkyPilot now supports running multiple isolated local API servers on one machine.

— States capability but no mechanism or setup steps v0.13.2rc1

10 **RunPod SSH key labeling** IMPROVED 30

Labels RunPod account SSH keys with the SkyPilot user.

— Minor named change, no further mechanism v0.13.2rc1

11 **Debug dump restricted to admins** BREAKING 25

Makes debug dump creation admin-only.

— Bare statement, no scope or migration guidance v0.13.2rc1

v0.13.2rc1 NOTES ↗ PRE-RELEASE

SkyPilot v0.13.2rc1 adds multi-server isolation, GCP federated credentials, richer Slurm visibility, and Kueue-aware provisioning timeouts.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.13.2rc1 https://github.com/skypilot-org/skypilot.git
# already have the repo? check out this version:
$ git checkout v0.13.2rc1
```

> Supports running multiple isolated local API servers on one machine.
>
> Supports GCP federated credentials for GCS `COPY` operations.
>
> Excludes Kueue admission wait time from `provision_timeout` so queue-gated pods are not incorrectly timed out.
>
> Adds SlurmClient.get_all_jobs_info() for per-node job attribution across Slurm clusters.
>
> Enriches Slurm node info with CPU/memory allocation and sampled usage metrics.
>
> +14 more

WAS THIS USEFUL? 👍 👎

Anthropic TypeScript SDK

Feedback ⓘ sdk-v0.127.0

NOTES ↗ RANK

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

> The Anthropic TypeScript SDK adds explicit context compaction control to the agentic tool runner and enriches rate-limit response metadata with a named group object.

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

01 **compactBeforeNextTurn on tool runner** NEW 80

The tool runner gains a compactBeforeNextTurn() method, letting callers trigger a context compaction step before the next agentic turn instead of waiting for automatic threshold logic. Example usage: call await runner.compactBeforeNextTurn() on a client.beta.messages.toolRunner() instance before iterating over its messages.

Trigger compaction explicitly before a long agentic turn to keep context within limits without waiting for automatic threshold logic.

typescript

```typescript
const runner = client.beta.messages.toolRunner({
  model: 'claude-opus-4-6',
  max_tokens: 4096,
  tools: [search, done],
  messages: [{ role: 'user', content: 'Research all ten topics.'
}],
});

// Compact before the next turn when you know context is growing
large
await runner.compactBeforeNextTurn();

for await (const message of runner) {
  console.log(message.usage.input_tokens);
}
```

— Names exact method and shows runnable usage example sdk-v0.127.0

THINNER COVERAGE BELOW

02 **Rate-limit response group field** DEPRECATED 45

Rate-limit API responses now include a `group` object with a `display_name` field; the previous `group_type` field is now deprecated.

— Names fields but gives no usage example or migration detail sdk-v0.127.0

sdk-v0.127.0 NOTES ↗ LATEST STABLE

Adds compactBeforeNextTurn() to the tool runner and a `group` field with `display_name` to rate-limit responses.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch sdk-v0.127.0
https://github.com/anthropics/anthropic-sdk-typescript.git
# already have the repo? check out this version:
$ git checkout sdk-v0.127.0
```

└──▷ **USE IT**

Trigger compaction explicitly before a long agentic turn to keep context within limits without waiting for automatic threshold logic.

typescript

```typescript
const runner = client.beta.messages.toolRunner({
  model: 'claude-opus-4-6',
  max_tokens: 4096,
  tools: [search, done],
  messages: [{ role: 'user', content: 'Research all ten topics.' }],
});

// Compact before the next turn when you know context is growing
large
await runner.compactBeforeNextTurn();

for await (const message of runner) {
  console.log(message.usage.input_tokens);
}
```

Adds compactBeforeNextTurn() method to the tool runner, letting callers trigger a context compaction step before the next agentic turn.

Adds `group` object with `display_name` field to rate-limit API responses; `group_type` is now deprecated.

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ v1.7.0 NOTES ↗ RANK

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

> The Anthropic Python SDK's v1.7.0 release adds a context-compaction trigger to the agentic tool runner and richer rate limit metadata.

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

01 **Manual context compaction in tool runner** NEW 60

Adds a compact_before_next_turn() method to the tool runner to trigger context compaction before the next agentic turn.

— Names exact method but no detail on compaction mechanism. v1.7.0

THINNER COVERAGE BELOW

02 **Rate limit group metadata** IMPROVED 45

Adds a `group` field with `display_name` to rate limit objects in the API response model.

— Names fields but not how they're used. v1.7.0

v1.7.0 NOTES ↗ LATEST STABLE

Adds compact_before_next_turn() to the tool runner and a `group` field with `display_name` to rate limits.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.7.0 https://github.com/anthropics/anthropic-sdk-python.git
# already have the repo? check out this version:
$ git checkout v1.7.0
```

Adds compact_before_next_turn() method to the tool runner to trigger context compaction before the next agentic turn.

Adds `group` with `display_name` field to rate limit objects in the API response model.

WAS THIS USEFUL? 👍 👎

Unsloth AI Unsloth

Feedback ⓘ v0.1.811-beta NOTES ↗

RANK

Unsloth runs and fine-tunes language and diffusion models locally through a Python library and interface.

> Unsloth added fast inference support for Qwen3.5 models and introduced detection for Ascend NPU devices.

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

01 **Fast inference support for Qwen3.5** NEW 40

Unsloth adds `fast_inference` support for Qwen3.5 models, enabling accelerated inference for this model family.

— Names the flag and model but no mechanism detail. v0.1.811-beta

02 **Ascend NPU device detection** NEW 25

Unsloth adds detection for Ascend NPU devices.

— Bare feature name, no further detail given. v0.1.811-beta

v0.1.811-beta NOTES ↗ PRE-RELEASE

Unsloth: Adds `fast_inference` support for Qwen3.5 models.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.811-beta
https://github.com/unslothai/unsloth.git
# already have the repo? check out this version:
$ git checkout v0.1.811-beta
```

Adds Ascend NPU device detection.

WAS THIS USEFUL? 👍 👎

Eigen Labs Darkbloom

Feedback ⓘ v0.9.6 NOTES ↗ CODE ↗

RANK

Private Inference Network on Idle Macs

> Darkbloom v0.9.6 adds native runtime support for the Bonsai 2 27B model and a small branding update to console capacity stats.

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

01 **Native Bonsai 2 27B runtime support** NEW 55

Adds native support for running the Bonsai 2 27B model, covering text, vision, and tool use, including paged KV cache. The runtime preserves the published checkpoint and its native precision.

— Names model and capabilities but no config or invocation details v0.9.6

02 **NVIDIA branding for Nemotron in stats view** IMPROVED 35

Adds NVIDIA branding for Nemotron models in the console stats capacity view.

— Minor cosmetic change, thinly described v0.9.6

v0.9.6 NOTES ↗ CODE ↗ LATEST STABLE

Darkbloom v0.9.6 adds native Bonsai 2 27B runtime with text, vision, and tool support including paged KV cache.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.9.6 https://github.com/Layr-Labs/d-inference.git
# already have the repo? check out this version:
$ git checkout v0.9.6
```

Adds native Bonsai 2 27B runtime support for text, vision, tools, and paged KV cache, preserving the published checkpoint and its native precision.

Adds NVIDIA branding for Nemotron models in the console stats capacity view.

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.16.0 NOTES ↗ RANK

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

> OpenAI Python SDK v3.16.0 adds programmatic webhook endpoint management to the client API.

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

01 **Webhook endpoint management via client API** NEW 35

The SDK now supports creating and managing webhook endpoints programmatically through the client, rather than requiring manual configuration.

— No method names, signatures, or code sample given v3.16.0

v3.16.0 NOTES ↗ LATEST STABLE

OpenAI Python SDK v3.16.0 adds webhook endpoint management via the client API.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v3.16.0 https://github.com/openai/openai-python.git
# already have the repo? check out this version:
$ git checkout v3.16.0
```

WAS THIS USEFUL? 👍 👎

Snowflake Cortex AI

Feedback ⓘ

Sep 16, 2026: Cortex Agents object enhancements (General availability) NOTES ↗

RANK

Snowflake Cortex AI provides managed language-model functions, search, and agents that work with data governed in Snowflake.

> Cortex Agents reached general availability with session-scoped temporary agents, a secure mode that hides agent specs from non-owner roles, privilege-preserving replacement via COPY GRANTS, and support for hosting agents in a user's Personal Database.

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

01 **Temporary agents for session-scoped testing** NEW 72

Adds `CREATE TEMPORARY AGENT` / `CREATE TEMP AGENT` to create agents scoped to the current session, intended for testing, one-off workflows, and ephemeral pipelines so no permanent objects are left behind.

Spin up a throwaway agent scoped to the current session for ad-hoc pipeline testing without leaving permanent objects behind.

```
$ CREATE TEMPORARY AGENT my_test_agent ...;
```

— Named command with runnable example, but no lifecycle/limit details

Sep 16, 2026: Cortex Agents object enha…

02 **COPY GRANTS on agent replacement** NEW 72

Adds `COPY GRANTS` support via `CREATE OR REPLACE AGENT ... COPY GRANTS` to preserve explicit privileges (except OWNERSHIP) when replacing an agent, so downstream roles retain access after a swap.

Replace an existing agent while carrying forward all explicitly granted privileges, so downstream roles do not lose access after the swap.

```
$ CREATE OR REPLACE AGENT my_agent COPY GRANTS ...;
```

— Exact syntax and runnable example, scope of preserved privileges named

Sep 16, 2026: Cortex Agents object enha…

THINNER COVERAGE BELOW

03 **Secure agents mode** NEW 45

Adds a secure agents mode that prevents roles able to invoke or modify an agent — but lacking the owner role — from retrieving the agent specification through metadata surfaces.

— Mechanism described but no enabling syntax or command shown

Sep 16, 2026: Cortex Agents object enha…

04 **Agents in Personal Database** NEW 37

Enables creating and running agents in the `PUBLIC` schema of a user-specific Personal Database.

— Names the schema and hosting location but gives no usage example

Sep 16, 2026: Cortex Agents object enha…

Sep 16, 2026: Cortex Agents object enhancements (General availability)

NOTES ↗ LATEST STABLE

Cortex Agents GA: temporary agents, secure agents, COPY GRANTS support, and Personal Database hosting

└──▷ **TRY IT**

Spin up a throwaway agent scoped to the current session for ad-hoc pipeline testing without leaving permanent objects behind.

```
$ CREATE TEMPORARY AGENT my_test_agent ...;
```

Replace an existing agent while carrying forward all explicitly granted privileges, so downstream roles do not lose access after the swap.

```
$ CREATE OR REPLACE AGENT my_agent COPY GRANTS ...;
```

Adds `CREATE TEMPORARY AGENT` / `CREATE TEMP AGENT` to create session-scoped agents for testing, one-off workflows, and ephemeral pipelines.

Adds secure agents mode, preventing roles that can invoke or modify an agent — but lack the owner role — from retrieving the agent specification through metadata surfaces.

Adds `COPY GRANTS` support via `CREATE OR REPLACE AGENT ... COPY GRANTS` to preserve explicit privileges (except OWNERSHIP) when replacing an agent.

Enables creating and running agents in the `PUBLIC` schema of a user-specific Personal Database.

WAS THIS USEFUL?

OpenRouter

Feedback ⓘ snapshot-20260919 NOTES ↗ RANK

OpenRouter is a routing gateway that exposes hundreds of models from many providers behind one OpenAI-compatible API with failover and unified billing.

> OpenRouter added image editing, letting users submit a source image with an edit prompt in one API call and swap the underlying model with a single field change.

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

01 **Image editing via unified API** NEW 45

OpenRouter now supports sending a source image plus an edit prompt in a single API request to receive an edited image back, and the image editing model can be swapped by changing a single request field.

— Describes behavior and model-switch mechanism but no field/endpoint names given.

snapshot-20260919

snapshot-20260919 NOTES ↗ LATEST STABLE

> OpenRouter adds image editing: send a source image plus a prompt in one request and swap models with one field change.
>
> Supports sending a source image and an edit prompt in a single API request to receive an edited image back.
>
> Enables switching the image editing model by changing a single request field.

WAS THIS USEFUL?

◆ Local LLM Runtimes

SGLang

Feedback ⓘ v0.5.20 NOTES ↗ CODE ↗ RANK

SGLang runs large language models through an inference server with efficient structured generation.

> SGLang's v0.5.20 release (713 PRs) headlines this window with sampling masks for RL rollout replay, a CPU-only scheduling simulator, a unified radix tree for SWA caching, and retirement of CUDA 12 and ROCm 7.0 support, while documentation updates add experimental NVFP4 KV-cache quantization for SM100, Ascend NPU quantization formats, a new `tlru` cache eviction policy, and prefill CUDA graph length capping.

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

01 NVFP4 KV-cache quantization on SM100 NEW 95

Adds experimental NVFP4 KV-cache support for SM100 GPUs via `--prefill-kv-cache-dequant-dtype` (default `auto`, picks native NVFP4 on SM100 and FP8 E4M3 on SM120), delivering a theoretical 3.56× KV-token capacity multiplier over BF16 (0.5625 vs 2 bytes/scalar). Supports two experimental E2M1 recipes — NVIDIA NVFP4 (16-value blocks, E4M3 block scales, per-tensor global scale) and an E2M1 block-size-16 compatibility recipe distinct from OCP MXFP4 block-size-32 — plus `--speculative-ngram-max-bfs-breadth=1` for breadth-1 NGRAM speculative decoding, Top-k-1 EAGLE/EAGLE3/NEXTN speculative decoding, and `--speculative-attention-mode` auto-resolved to `decode` in mixed FlashInfer-prefill/TRT-LLM-decode recipes.

Serve an FP8 model on SM100 using native NVFP4 KV-cache for both prefill and decode, maximizing KV-token capacity (up to 3.56× vs BF16).

```
$ python -m sglang.launch_server --model-path Qwen/Qwen3.5-35B-A3B-FP8 --tp-size 4 --prefill-kv-cache-dequant-dtype auto --page-size 16
```

Run the mixed recipe where prefill dequantizes NVFP4 into an FP8 E4M3 workspace while decode still consumes native NVFP4 — useful when native NVFP4 prefill constraints (head dim divisible by 64) cannot be met.

```
$ python -m sglang.launch_server --model-path Qwen/Qwen3.5-35B-A3B-FP8 --tp-size 4 --prefill-kv-cache-dequant-dtype fp8_e4m3 --page-size 16
```

Enable NGRAM speculative decoding with breadth-1 on the native NVFP4 path to comply with the SM100 NVFP4 speculative decoding requirement.

```
$ python -m sglang.launch_server --model-path Qwen/Qwen3.5-35B-A3B-FP8 --tp-size 4 --prefill-kv-cache-dequant-dtype auto --page-size 16 --speculative-ngram-max-bfs-breadth=1
```

— Full mechanism, named flags and runnable launch commands given. product docs

02 **Prefill CUDA graph max sequence length cap** NEW 90

Adds `--cuda-graph-max-seq-len-prefill` flag to cap the longest sequence a prefill CUDA graph replay admits; longer batches fall back to eager prefill, configurable via `cuda_graph_config[prefill].max_seq_len`.

Limit prefill CUDA graph replay to sequences up to 2048 tokens so longer prompts run eager rather than being rejected.

```
$ python -m sglang.launch_server --cuda-graph-max-seq-len-prefill 2048 --model-path <model>
```

— Named flag and config key with a runnable launch example. product docs

03 **Sampling masks for RL rollout replay** NEW 90

Adds `--sampling-mask-max-tokens` (default 4096) to cap sampling-mask capacity, and a per-request `return_sampling_mask` flag to return the exact token support and log-probability at each decode step for RL rollout replay without reconstructing top-k or top-p; masks run under overlap scheduling, yielding 17% higher decode throughput at batch 1 and 52% at batch 64 on Qwen3-8B. Also adds `DisallowedTokensLogitsProcessor` support alongside sampling masks.

— Named flags with concrete throughput numbers, no runnable example. v0.5.20

04 **Opt-in retention for /v1/responses** BREAKING 90

Adds `--enable-response-store` server flag to opt in to `/v1/responses` result retention; without it, retrieval, `previous_response_id` chaining, and background requests return HTTP 400. This is a behavior change from prior default: existing setups relying on retrieval or chaining must restart the server with `--enable-response-store` (not available for PD deployments).

Enable `/v1/responses` result retention so an RL trainer can retrieve prior responses and chain them with `previous_response_id`.

```
$ python -m sglang.launch_server --model <model> --enable-response-store
```

— Named flag, endpoint, HTTP status and runnable command. v0.5.20

05 **FlashInfer MegaMoE backend for DeepSeek-V4-Flash** NEW

90

Adds `--moe-runner-backend flashinfer_megamoe` to enable FlashInfer MegaMoE on DeepSeek-V4-Flash NVFP4 at TP4/DP4, gaining up to 11.9% prefill throughput at 8192 tokens per rank.

Launch a server with the FlashInfer MegaMoE backend to gain up to 11.9% prefill throughput on DeepSeek-V4-Flash NVFP4 at TP4/DP4 on Blackwell.

```
$ python -m sglang.launch_server --model deepseek-ai/DeepSeek-V4-Flash --moe-runner-backend flashinfer_megamoe --tp 4 --dp 4
```

— Named flag, scope and throughput gain with runnable example. v0.5.20

06 **Tail-Optimized LRU radix cache eviction policy** NEW

85

Adds `tlru` (Tail-Optimized LRU, arXiv:2510.15152) as a new radix cache eviction policy that protects only the cached history a conversation's next prefill needs to stay under a TTFT budget, configured via `threshold` (tail-latency threshold ξ, in tokens) and `next_prompt_estimate` (estimated tokens the next turn will add) as JSON, e.g. `{"threshold": 4096, "next_prompt_estimate": 512}`.

— Names config parameters with example JSON, no full launch command. product docs

07 **Unified radix tree with branching-point caching for SWA** NEW

80

Introduces a unified radix tree with branching-point caching for the SWA component, raising token hit rate from 43.8% to 60.8% and dropping mean TTFT from 1.57s to 1.07s on DeepSeek-V4-Flash with a shared system prompt; adds an opt-in external linker for the unified radix tree to address a shared global memory pool through Mooncake or UMBP.

— Detailed before/after metrics and named backing systems, no config key given. v0.5.20

08 **Ascend 950PR/DT NPU quantization formats** NEW

75

Adds MXFP4 quantization (W4A4, MXFP4 weights+activations) for Qwen3 dense and MoE LLMs on Ascend 950PR/DT Series NPUs, with online dual-level MXFP4 for dense

models; MXFP8 quantization for LLM Dense Linear and LLM MoE (FusedMoE, e.g. Qwen3-30B-A3B), supporting online and offline modes via CANN; W4A8 quantization (MXFP4 weights + MXFP8 activations) for Qwen3 dense LLMs, online mode; and MXFP8 quantization for diffusion models (e.g. Wan2.2) via SGLang-Diffusion, supporting online and offline (ModelSlim) modes with CANN >= 8.0.RC3.

— Names formats, models and CANN requirements but no exact flags. product docs

09 **TRT-LLM attention kernels for DeepSeek-V4 on Blackwell**

IMPROVED 75

Adds TRT-LLM attention kernels covering DeepSeek-V4's CSA and HCA layers on SM100 and SM103 (Blackwell B200), delivering ~1.2x faster prefill and ~1.45x faster decode than FlashMLA at the kernel level.

— Concrete kernel scope and speedups, no flag to invoke it. v0.5.20

10 **CPU-only SGLang Simulator for scheduling studies** NEW

70

Adds an SGLang Simulator, a CPU-only tool that runs the real scheduler, radix cache, and hierarchical cache with a latency predictor in place of the model forward pass, predicting TTFT within ~6% and prefix reuse within 0.05 percentage points — enabling cache and scheduling studies without GPUs.

— Explains mechanism and accuracy numbers but no invocation shown. v0.5.20

11 **DSpark draft KV transfer for disaggregated hybrid serving**

NEW 70

Enables DSpark draft KV transfer from a DCP1 prefill to a DCP-N decode under disaggregated, context-parallel serving for hybrid models such as Kimi-Linear, verified on 8x B300 over NIXL and Mooncake up to 256K input.

— Names mechanism, hardware and scale, no invocation given. v0.5.20

12 **DeepGEMM paged-MQA and FP4 MoE backend on SM120** IMPROVED

70

Enables the DeepGEMM paged-MQA kernel and DeepGEMM FP4 MoE backend on SM120 (RTX PRO 6000) for DeepSeek-V4, dropping decode TPOT from 36.1 to 10.5 ms at batch 1 and reducing TTFT by 20% from 8K to 128K input.

— Precise before/after metrics, but no flag to enable it. v0.5.20

13 **New release images for XPU, ROCm 10, gfx1151, MUSA** NEW

65

Adds Intel XPU versioned release images (`lmsysorg/sglang:vX.Y.Z-xpu`) built from the XPU Dockerfile on every tagged release, ROCm 10 release images and kernel wheel

for MI30x and MI35x, a gfx1151 Docker image for Strix Halo/Ryzen AI MAX+, and Moore Threads MUSA release images.

— Names every image and hardware target, no pull commands given. v0.5.20

14 **ROCm model loading speedup** IMPROVED 65

Speeds up model loading on ROCm by staging large pageable host-to-device copies instead of pinning them in place; GLM-5.2 at TP4 on 4x MI355X now loads in 40.4s instead of 505.7s.

— Concrete before/after benchmark, no flag or setting named. v0.5.20

15 **CUDA 12 and ROCm 7.0 support retirement** DEPRECATED 65

Retires the CUDA 12 lane — v0.5.19 was the last release with `-cu12x` wheels and images, so CUDA 12 users must migrate — and retires ROCm 7.0 CI, images, and kernel wheel, meaning ROCm 7.0 deployments will no longer receive images or wheel support.

— Clear migration cutoff versions named for both platforms. v0.5.20

THINNER COVERAGE BELOW

16 **New model support across LLM and diffusion** NEW 50

Adds native serving support for new autoregressive models GLM-5.3-Flash, Hy4-Preview, Qwen3.8-Flash-Next, K2 Horizon, and Nanbeige4.2, plus diffusion models SenseNova-U1.5-8B-MoT, FastH3, and VDN-H3.

— Names all new models but gives no config or usage detail. v0.5.20

17 **Profiling and metrics additions** NEW 45

Adds `[Profiler]` labels DRAFT and VERIFY to draft-runner and target-verify steps in speculative decoding step spans, and reports logical prefill token counts in metrics.

— Names the labels and metric but no access path. v0.5.20

18 **SGLang Diffusion Prometheus metrics** NEW 40

Exposes Prometheus metrics for SGLang Diffusion covering requests, queues, pipeline stages, and LoRA adapter state.

— Lists metric categories but no metric names or endpoint. product docs

19 **DeepEP dispatcher output dtype selection on Ascend** NEW 35

Adds a new option to select the DeepEP dispatcher output dtype, supported only on Ascend A2/A3 Series NPUs.

— Bare mention of an option with no name or config path given. product docs

20 **Speculative decoding policy enhancements** NEW 30

Adds native UNO serving support for speculative decoding and support for a custom policy for adaptive speculative decoding.

— Two thin bullets with no flags or mechanism named. v0.5.20

└──▷ **BREAKING ON UPGRADE**

! The CUDA 12 lane is retired; v0.5.19 was the last release with `-cu12x` wheels and images — CUDA 12 users must migrate.

! ROCm 7.0 CI, images, and kernel wheel are retired; ROCm 7.0 deployments will no longer receive images or wheel support.

! `/v1/responses` no longer retains results in memory by default; existing setups relying on retrieval, `previous_response_id` chaining, or background requests will receive HTTP 400 unless the server is restarted with `--enable-response-store`. PD deployments cannot enable it.

v0.5.20 NOTES ↗ CODE ↗ LATEST STABLE

SGLang v0.5.20 adds sampling masks for RL rollouts, a CPU simulator, unified radix tree, opt-in response storage, and Intel XPU release images across 713 PRs.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.5.20 https://github.com/sgl-
project/sglang.git
# already have the repo? check out this version:
$ git checkout v0.5.20
```

└──▷ **TRY IT**

Launch a server with the FlashInfer MegaMoE backend to gain up to 11.9% prefill throughput on DeepSeek-V4-Flash NVFP4 at TP4/DP4 on Blackwell.

```
$ python -m sglang.launch_server --model deepseek-ai/DeepSeek-V4-
Flash --moe-runner-backend flashinfer_megamoe --tp 4 --dp 4
```

Enable `/v1/responses` result retention so an RL trainer can retrieve prior responses and chain them with `previous_response_id`.

```
$ python -m sglang.launch_server --model <model> --enable-response-
store
```

Adds `--sampling-mask-max-tokens` (default 4096) to cap sampling-mask capacity, and `return_sampling_mask` per-request to return the exact token support and log-

probability at each decode step for RL rollout replay without reconstructing top-k or top-p; masks run under overlap scheduling, yielding 17% higher decode throughput at batch 1 and 52% at batch 64 on Qwen3-8B.

Adds `--enable-response-store` server flag to opt in to `/v1/responses` result retention; without it, retrieval, `previous_response_id` chaining, and background requests return HTTP 400.

Adds `--moe-runner-backend flashinfer_megamoe` to enable FlashInfer MegaMoE on DeepSeek-V4-Flash NVFP4 at TP4/DP4, gaining up to 11.9% prefill throughput at 8192 tokens per rank.

Adds an SGLang Simulator: a CPU-only tool that runs the real scheduler, radix cache, and hierarchical cache with a latency predictor in place of the model forward, predicting TTFT within ~6% and prefix reuse within 0.05 percentage points for cache and scheduling studies without GPUs.

Adds native UNO serving support for speculative decoding.

+15 more

└─▷ **BREAKING ON UPGRADE**

- ! The CUDA 12 lane is retired; v0.5.19 was the last release with `-cu12x` wheels and images — CUDA 12 users must migrate.
- ! ROCm 7.0 CI, images, and kernel wheel are retired; ROCm 7.0 deployments will no longer receive images or wheel support.
- ! `/v1/responses` no longer retains results in memory by default; existing setups relying on retrieval, `previous_response_id` chaining, or background requests will receive HTTP 400 unless the server is restarted with `--enable-response-store`. PD deployments cannot enable it.

WAS THIS USEFUL? 👍 👎

vMLX

Feedback ⓘ v1.6.63 NOTES ↗ RANK

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

> vMLX shipped an FP16 activation and KV-cache precision policy for Bonsai 2 27B JANG alongside 1.75-bit packed weight storage, but the change breaks compatibility with existing FP32 cache states.

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

01 FP16 precision policy for Bonsai 2 27B JANG

BREAKING

73

Bonsai 2 27B JANG now keeps language activations and attention KV cache in FP16, while Hadamard accumulation, checkpoint parameters, native recurrent state, and the vision tower retain their required precision; the model also gains support for 1.75-bit packed weight storage. Because prefix and SSD cache identity now includes the activation precision policy, existing FP32 cache states cannot be restored into the new FP16 execution path — existing model bundles and cache files are not rewritten and must be regenerated.

— Mechanism and scope named, but no migration command given v1.6.63

└──▷ **BREAKING ON UPGRADE**

! Prefix and SSD cache identity now includes the activation precision policy — existing FP32 cache states cannot be restored into the changed FP16 execution path. Existing model bundles and cache files are not rewritten and must be regenerated.

v1.6.63 NOTES ↗ LATEST STABLE

Bonsai 2 27B JANG gains FP16 activation/KV precision policy with 1.75-bit packed weight storage support

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.6.63 https://github.com/jjang-ai/vmlx.git
# already have the repo? check out this version:
$ git checkout v1.6.63
```

Bonsai 2 27B JANG now keeps language activations and attention KV cache in FP16, with Hadamard accumulation, checkpoint parameters, native recurrent state, and the vision tower retaining their required precision.

└──▷ **BREAKING ON UPGRADE**

! Prefix and SSD cache identity now includes the activation precision policy — existing FP32 cache states cannot be restored into the changed FP16 execution path. Existing model bundles and cache files are not rewritten and must be regenerated.

WAS THIS USEFUL? 👍 👎

LM Studio

Feedback ⓘ

Splash Engine - the fastest local Qwen3.8 on Apple Silicon (2026-09-18) BLOG ↗

RANK

LM Studio is a desktop application that runs local LLMs and provides model discovery, downloads, chat, and a local API server.

> LM Studio Bionic ships an experimental Metal-based inference engine, Splash, delivering dramatically faster local inference for Qwen3-series models on Apple Silicon.

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

01 Splash (Metal) inference engine for Apple Silicon NEW

93

A new experimental backend, Splash (Metal), is downloadable via `Settings > Runtime` under Experimental backends and delivers up to ~74 tokens/s on short prompts and 54 tokens/s at 32K context on an M5 Pro with 48 GB unified memory, running on M3-or-newer Macs with at least 36 GB unified memory. It supports `incoai/Qwen3.8-27B-Splash` and `incoai/Qwen3.6-35B-A3B-Splash`, downloadable via `Settings > Explore`, each bundled with a dedicated DFlash 2 draft model for speculative decoding, and achieves 170 tokens/s combined throughput with four concurrent requests on short prompts — 3.9× the next-fastest engine measured on Qwen3.8-27B.

Install the Splash (Metal) backend and run Qwen3.8-27B locally at maximum speed on an M3-or-newer Mac with at least 36 GB unified memory.

📍 1. Open LM Studio Bionic 1.1.5 or newer. 2. Go to Settings › Runtime. 3. Under Experimental backends, click Download next to Splash (Metal). 4. Go to Settings › Explore and paste `incoai/Qwen3.8-27B-Splash` into the search bar. 5. Select the model and click Download. 6. Start a new session and select `incoai/Qwen3.8-27B-Splash` from the local model picker.

— Names exact backend, models, settings paths, and benchmark numbers

Splash Engine - the fastest local Qwen3…

> Splash Engine - the fastest local Qwen3.8 on Apple Silicon (2026-09-18)
>
> BLOG ↗ LATEST STABLE
>
> LM Studio Bionic adds Splash Engine (Metal) backend for up to 74 tok/s local inference on Qwen3.8-27B on Apple Silicon.

└──▷ **HOW TO FIND IT**

Install the Splash (Metal) backend and run Qwen3.8-27B locally at maximum speed on an M3-or-newer Mac with at least 36 GB unified memory.

📍 1. Open LM Studio Bionic 1.1.5 or newer. 2. Go to Settings › Runtime. 3. Under Experimental backends, click Download next to Splash (Metal). 4. Go to Settings › Explore and paste `incoai/Qwen3.8-27B-Splash` into the search bar. 5. Select the model and click Download. 6. Start a new session and select `incoai/Qwen3.8-27B-Splash` from the local model picker.

Adds the Splash (Metal) inference engine as an experimental backend, downloadable via Settings > Runtime, delivering ~74 tokens/s on short prompts and 54 tokens/s at 32K context on an M5 Pro with 48 GB unified memory.

Supports `incoai/Qwen3.8-27B-Splash` and `incoai/Qwen3.6-35B-A3B-Splash` as locally runnable models via Settings > Explore, each bundled with a dedicated DFlash 2 draft model for speculative decoding.

Achieves 170 tokens/s combined throughput with four concurrent requests on short prompts — 3.9× the next-fastest engine measured on Qwen3.8-27B.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20260919 NOTES ↗

RANK

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

> LangSmith's latest snapshot adds a `GET /v2/threads/stats` endpoint with streaming support, IAM role assumption for private ECR registries, new OAuth authorization parameters for sharing managed credentials, and tightens dataset export permissions in a breaking change.

└──▷ **WHAT SHIPPED · 16 FEATURES** most completely described first ? what's the number?

01 Thread statistics endpoint with streaming support NEW

91

Adds the `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. Thread query clients can now request server-sent events on `/v2/threads/stats` to receive results progressively while preserving cursor pagination.

Pull thread-level latency, token usage, and cost statistics for a tracing project in a single API call.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

— Named endpoint with method, path and runnable curl example. snapshot-20260919

02 Configurable license expiration warning window NEW

86

Adds `LICENSE_EXPIRATION_WARNING_DAYS` environment variable for self-hosted operators to configure how many days before license expiration the warning banner appears (default: 7 days).

Set the license-expiration warning horizon to 14 days on a self-hosted deployment so the ops team gets earlier notice.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

– Named env var with default and runnable example. snapshot-20260919

03 **Gateway-backed prompt manifest standardization** IMPROVED 76

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, compatible with standard LangChain SDKs.

– Names manifest type, URL, and secret key used. snapshot-20260919

04 **ls_provider field for Gateway model configs** NEW 75

Adds `ls_provider` field on OpenAI-compatible model configurations in LLM Gateway to populate trace metadata and match custom models to provider-specific pricing.

Route a custom OpenAI-compatible model through the LLM Gateway and tie its traces to provider-specific pricing by setting `ls_provider`.

```json
{
  "model": "my-custom-model",
  "base_url": "https://gateway.langsmith.com/oai/v1",
  "ls_provider": "openai",
  "api_key": "${LANGSMITH_GATEWAY_API_KEY}"
}
```

– Named config field with example, but limited mechanism detail. snapshot-20260919

05 **OAuth authorization session owner parameters** NEW 65

OAuth authorization sessions now accept `owner_type` and `owner_id` parameters, allowing a human to authorize a managed credential for either a user or an agent; a separate `owner_id` parameter for workspace owners lets one managed OAuth credential be shared across a workspace.

– Named parameters but no endpoint or code sample given. snapshot-20260919

06 **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, breaking existing API integrations that rely on read permission alone.

– Clear before/after and migration implication for API callers. snapshot-20260919

07 **Whole-thread dataset and annotation queue actions** NEW 63

Adds Add to dataset and Add to annotation queue buttons to the thread header in every thread view (Messages, Turns, and Details), enabling whole-thread addition without selecting a run. The add-to-dataset popup for threads also gained a New Dataset option that creates the dataset empty and adds the selected threads to it.

— UI paths named across thread views, no code surface. snapshot-20260919

08 **Bedrock Playground timeout and retry options** IMPROVED 60

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.

— Gives before/after limit but no exact UI path. snapshot-20260919

THINNER COVERAGE BELOW

09 **LLM Gateway fallback timeout handling** IMPROVED 58

LLM Gateway fallback chains can now advance to the next model when a candidate does not return response headers within a configured timeout, while active response streams continue after headers arrive.

— Describes behavior change but no config knob named. snapshot-20260919

10 **IAM role assumption for private ECR registries** IMPROVED 54

Commercial private ECR registries can now assume an AWS IAM role on demand, enabling repository and tag search without expiring saved authorization tokens.

— Explains mechanism but no config surface named. snapshot-20260919

11 **Trajectory batch reads on dedicated gRPC service** IMPROVED 45

Trajectory batch reads now run on a dedicated in-cluster gRPC service instead of the smith-go monolith, isolating trajectory evaluation load from UI and API traffic.

— Architectural change explained but not user-actionable. snapshot-20260919

12 **MCP connectors adopt Client ID Metadata Documents** IMPROVED 43

LangSmith MCP connectors now use Client ID Metadata Documents when supported, with dynamic client registration retained as a compatibility fallback.

— Names the mechanism but no config steps given. snapshot-20260919

13 **Claude Fable 5.1 model availability** NEW 43

Claude Fable 5.1 is now available in the Anthropic, Bedrock, and Vertex AI model selectors in the Playground.

— Names model and selectors but no further detail. snapshot-20260919

14 **IssueBench Opus 5 regression benchmark** NEW 38

IssueBench now includes a benchmark task that compares pre- and post-Opus 5 agent traces for cost and trajectory regressions.

— Names the benchmark but no usage detail. snapshot-20260919

15 **Workspace switcher accepts workspace IDs** IMPROVED 30

The workspace switcher now accepts workspace IDs in addition to workspace names.

— Very thin, single-line description. snapshot-20260919

16 **Agent OAuth connection result page** IMPROVED 30

Agent OAuth connections now show a provider-branded result page after authorization, distinguishing successful from failed connection attempts.

— UI change described without navigation or mechanism detail. snapshot-20260919

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read and download permissions; existing API integrations that rely on dataset read permission alone will receive a forbidden response.

snapshot-20260919 NOTES ↗ LATEST STABLE

LangSmith adds `/v2` /threads/stats API, IAM role assumption for ECR, Gateway fallback timeouts, and `LICENSE_EXPIRATION_WARNING_DAYS` config

└──▷ TRY IT

Pull thread-level latency, token usage, and cost statistics for a tracing project in a single API call.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

Route a custom OpenAI-compatible model through the LLM Gateway and tie its traces to provider-specific pricing by setting `ls_provider`.

```json
{
  "model": "my-custom-model",
  "base_url": "https://gateway.langsmith.com/oai/v1",
  "ls_provider": "openai",
  "api_key": "${LANGSMITH_GATEWAY_API_KEY}"
}
```

Set the license-expiration warning horizon to 14 days on a self-hosted deployment so the ops team gets earlier notice.

```.env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Adds `LICENSE_EXPIRATION_WARNING_DAYS` environment variable for self-hosted operators to configure how many days before license expiration the warning banner appears (default: 7 days).

Adds `ls_provider` field on OpenAI-compatible model configurations in LLM Gateway to populate trace metadata and match custom models to provider-specific pricing.

Adds the `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.

Thread query clients can now request server-sent events on `/v2/threads/stats` to receive results progressively while preserving cursor pagination.

OAuth authorization sessions now accept `owner_type` and `owner_id` parameters, allowing a human to authorize a managed credential for either a user or an agent.

+14 more

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read and download permissions; existing API integrations that rely on dataset read permission alone will receive a forbidden response.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.71 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik improved SDK setup and hardened backend readiness checks around span topology, while also speeding up bulk experiment uploads and adding experiment naming from the Playground.

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

01 **Readiness check for span topology consistency** IMPROVED — 50

The backend now fails readiness checks when the spans wrap flag disagrees with the database topology, preventing misconfigured deployments from accepting traffic.

– Explains the mechanism but no config key or flag name given 2.2.71

02 **Streamed bulk experiment uploads** IMPROVED — 45

Experiment bulk uploads are now streamed instead of requiring four full passes, improving upload performance for large experiment sets.

– Clear before/after but no runnable command or config 2.2.70

03 **Improved SDK configuration flow** IMPROVED — 25

The `opik` SDK configuration flow was improved for easier setup.

– Names the SDK but no detail on what changed 2.2.71

04 **Experiment naming from the Playground** NEW — 20

Users can now name experiments directly from the Playground.

– Only mentioned in summary, no further detail given 2.2.70

2.2.71 NOTES ↗ LATEST STABLE

Opik 2.2.71 improves the SDK configuration flow and adds backend readiness checks for span topology consistency.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch 2.2.71 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.71
```

Improves the `opik` SDK configuration flow for easier setup.

Fails readiness checks when the spans wrap flag disagrees with the database topology, preventing misconfigured deployments from accepting traffic.

2.2.70 NOTES ↗ STABLE

Opik 2.2.70 lets users name experiments from the Playground and streams bulk experiment uploads for faster ingestion.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 2.2.70 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.70
```

Streams experiment bulk uploads instead of four full passes, improving upload performance for large experiment sets.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20260919 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust expanded AI provider governance with project-scoped providers, model restrictions, and gateway routing headers, added TypeSafe Jev as a judge/classifier with call tracing, and introduced SAML/OIDC SSO with just-in-time domain provisioning, alongside new `bt` CLI observability tooling, a metrics columnstore, and billing analytics.

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

01 Project-scoped AI providers and gateway routing controls NEW

95

Adds project-scoped AI provider configurations, allowing separate credentials, billing, rate limits, or regional endpoints per project, independent of organization-level providers. Adds the `x-bt-endpoint-name` request header to the AI gateway to pin a request to a specific provider by its configuration name (e.g. `OPENAI_API_KEY` or a custom provider name). Adds `Registry models` restriction settings (`All models` , `Selected models` , `None`) on provider configurations to control which models a provider exposes. Adds a `Models` tab / `Browse and test models` panel on provider configurations with a `curl` command generator that inserts a Braintrust API key and the correct `x-bt-endpoint-name` header for immediate testing. Adds `Custom model` entries on provider `Setup` for models outside Braintrust's registry. Introduces organization-level and project-level provider permission scopes so Enterprise teams can combine model restrictions with provider permissions.

Force a gateway request to use a specific named provider — useful when multiple providers support the same model and you need to target one for billing isolation or regional routing.

```
$ curl https://api.braintrust.dev/v1/proxy/chat/completions \
  -H 'Authorization: Bearer $BRAINTRUST_API_KEY' \
  -H 'x-bt-endpoint-name: OPENAI_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gpt-4o", "messages": [{"role": "user",
"content": "Hello"}]}'
```

Lock a project provider to a specific subset of approved models so that team members cannot use models outside that set through the gateway.

📍 In your project, navigate to Settings › AI Providers. Click the provider row, select Setup, expand More settings, open Models, set Registry models to `Selected`

`models`, check each approved model, then click Save.

Restrict a project provider to only serve specific models so teams cannot access models outside the approved list.

📍 In the Braintrust console, go to Settings › AI Providers › [your provider] › Selected models and choose the approved model list for the project.

— Names header, config states, curl flow and UI path precisely `snapshot-20260919`

---

02 **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 to `BraintrustConfig` for tuning S3 attachment uploader behavior.

Cap the in-memory upload queue and set a request timeout so the S3 attachment uploader does not buffer unboundedly in high-throughput pipelines.

```
$ BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE=500
BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS=10000
BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES=5
BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS=200
python my_eval.py
```

— Four exact env vars with a runnable example command `product docs`

---

03 **TypeSafe Jev as judge and classifier with tracing** NEW

67

Integrates TypeSafe Jev, a model that returns a structured choice or numeric result rather than free text, usable as a judge in scorers and classifiers to score outputs against predefined criteria or label logs with a fixed category set, using a built-in model or a TypeSafe key. Braintrust records per-choice probabilities and confidence scores, and enables tracing of Jev calls from your application so you can inspect context, question criteria, and selected answers together to identify ambiguous decisions. Requires Python v0.41.0+ or TypeScript v3.34.0+, and data plane v2.15.0+ for use as a judge in scorers/classifiers.

— Mechanism and version requirements given, no runnable snippet `snapshot-20260919`

---

04 **SAML/OIDC SSO with just-in-time domain provisioning** NEW

65

Adds SAML and OIDC SSO support with just-in-time (JIT) provisioning via domain mappings: users from a configured email domain are automatically added to your Braintrust organization on first sign-in without requiring an invitation. Each domain mapping can optionally require a specific SAML group value, sent as a separate attribute by the IdP, to gate organization access, and can optionally assign the user to one Braintrust permission group on join.

— Mechanism explained but setup steps not detailed `product docs`

THINNER COVERAGE BELOW

05 **Project provider override scope narrowed on hosted deployments**

55

BREAKING

On Braintrust-hosted deployments, project providers now override organization providers only for the models they serve; organization providers may still serve other models using their own credentials. Teams that relied on project providers to restrict model access or spending across all models should review their organization provider model selections and permissions.

— Clear before/after but no exact config steps given `snapshot-20260919`

06 **bt CLI capture of Grok and Google Antigravity sessions** NEW

55

Adds the `bt` CLI (v0.19.3+) to capture Grok and Google Antigravity coding sessions as Braintrust traces across prompts, model calls, and tool activity.

— Names CLI version and tools but no command shown `snapshot-20260919`

07 **Metric columnstore for trace metrics** NEW

55

Adds Metric columnstore, enabled in project settings, which stores trace-level metrics in columnar format to reduce data read on log queries; requires data plane v2.2.1+.

— Explains benefit and requirement, UI path only `snapshot-20260919`

08 **Per-project usage analytics breakdown** NEW

55

Adds per-project breakdown to the Usage analytics section of the billing page, covering logs, scores, and Topics token spend, exportable as a CSV; requires data plane v2.13.0+.

— Names export format and page location, no exact steps `snapshot-20260919`

09 **DeepSeek V4.1 Flash model support** NEW 50

Adds `deepseek-v4.1-flash` as a routable model through the Braintrust Gateway with no AI provider setup required, and as a built-in model in playgrounds, prompts, and scorers — both with multimodal reasoning support.

— Names model and surfaces but no usage command given `snapshot-20260919`

10 **Java SDK 0.3.24 released** NEW 20

Releases Java SDK version 0.3.24.

— Bare version bump with no change details `product docs`

└─▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for the models they serve; organization providers may serve other models using their own credentials. If you relied on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

`snapshot-20260919` NOTES ↗ LATEST STABLE

Braintrust adds Patterns auto-investigation, Debugger, Loop automations, TypeSafe Jev tracing, observability templates, and metric columnstore.

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 `deepseek-v4.1-flash` as a routable model through the Braintrust Gateway with no AI provider setup required, supporting multimodal reasoning.

Adds the `bt` CLI (v0.19.3+) to capture Grok and Google Antigravity coding sessions as Braintrust traces across prompts, model calls, and tool activity.

Adds Metric columnstore, enabled in project settings, which stores trace-level metrics in columnar format to reduce data read on log queries; requires data plane v2.2.1+.

Adds per-project breakdown to the Usage analytics section of the billing page, covering logs, scores, and Topics token spend, exportable as a CSV; requires data plane v2.13.0+.

+3 more

└─▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for the models they serve; organization providers may serve other models using their own credentials. If you relied on project providers to restrict

model access or spending across all models, review organization provider model selections and permissions.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

ToolHive

Feedback ⓘ v0.50.0 NOTES ↗ RANK

ToolHive runs MCP servers in isolated containers and enforces identity and access policies for local and Kubernetes deployments.

> ToolHive v0.50.0 strengthens its authentication and plugin security surfaces, adding standards-based OAuth flows (RFC 8707 resource indicators, RFC 8628 device authorization grant), ID-JAG assertion support, catalog provenance enforcement for plugins, key rotation workflows, and resilience fixes for OIDC token refresh and non-conforming MCP backends.

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

01 **Tolerant handling of optional capability calls** IMPROVED 60

Tolerates `MethodNotFound` responses for optional capabilities in `ListCapabilities`, avoiding hard failures against non-conforming backends.

— Names exact method and response code involved v0.50.0

THINNER COVERAGE BELOW

02 **RFC 8628 device authorization grant** NEW 50

Adds RFC 8628 device authorization grant support, including a device flow verification page and DCR (dynamic client registration) support.

— Names the RFC and components, no exact endpoint or command v0.50.0

03 **RFC 8707 resource indicators in auth server** NEW 45

Adds support for RFC 8707 resource indicators as additional token-request params in the auth server.

— Names the RFC and param type but no exact API shape v0.50.0

04 **Catalog provenance enforcement for plugin installs** NEW 45

Enforces catalog provenance for plugin installs, requiring plugins to pass provenance checks before installation.

— Describes the enforcement but not the check mechanism v0.50.0

05 **OIDC refresh token validation retry** IMPROVED 45

Retries ID-token validation after OIDC refresh, recovering sessions that would otherwise fail on token expiry.

— Explains before/after behavior but no config surface v0.50.0

06 **ID-JAG assertion support in auth server** NEW 35

Adds acceptance of ID-JAG assertions with a bound JWT-bearer handler in the auth server.

— Mechanism named but no usage detail given v0.50.0

07 **Key rotation workflows for skills and plugins** NEW 35

Adds lock-driven skill key rotation and plugin key rotation workflows.

— Names two rotation types, no operational detail v0.50.0

v0.50.0 NOTES ↗ LATEST STABLE

ToolHive v0.50.0 adds RFC 8707 resource indicators, RFC 8628 device authorization grant, ID-JAG assertion support, and catalog provenance enforcement for plugin installs.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.50.0
https://github.com/stacklok/toolhive.git
# already have the repo? check out this version:
$ git checkout v0.50.0
```

Adds support for RFC 8707 resource indicators as additional token-request params in the auth server.

Adds RFC 8628 device authorization grant support, including a device flow verification page and DCR support.

Adds acceptance of ID-JAG assertions with a bound JWT-bearer handler in the auth server.

Enforces catalog provenance for plugin installs, requiring plugins to pass provenance checks before installation.

Adds lock-driven skill key rotation and plugin key rotation workflows.

+2 more

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 19, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue