THE AI TOOLCHAIN

NO. 035

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 23, 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 361 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 59 tools

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

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

Three unrelated things stand out. Pipelock now scans MCP traffic in-flight — tool-description drift, argument DLP, shell deobfuscation —

across six IDEs, so a swapped tool definition fails before the agent acts on it. LLM Compressor's Triton GPTQ kernel is roughly 15x faster, cutting quantization from an overnight job. Together AI adds canary and blue-green rollouts to model endpoints.

GOVERN

Catch a hijacked MCP tool before the agent calls it, not in the postmortem

Pipelock proxies MCP traffic and inspects it live — flagging tool descriptions that changed since approval, scanning arguments for exfiltrated secrets, deobfuscating shell payloads and binding sessions — with signed evidence bundles and SARIF for CI gating. Cursor turns Security Agent findings into a triage queue with one-click Cloud Agent fixes, and Copilot CLI extends managed-settings enforcement to ACP, AHP-host and server-mode sessions, closing the paths that previously ran outside admin policy.

Pipelock · Cursor · GitHub Copilot CLI

DEPLOY

Quantize a large MoE in an afternoon and ship the new weights behind a canary

LLM Compressor's Triton GPTQ kernel runs roughly 15x faster and REAP expert pruning drops whole MoE experts rather than just narrowing weights, so compressing a frontier-size model stops being a multi-day batch job. Together AI's canary, blue-green and rolling strategies via CLI and SDK mean the resulting endpoint gets real traffic in slices — a quality regression hits 5% of requests instead of all of them — with idle-timeout auto-shutdown reclaiming the GPU when it goes quiet.

LLM Compressor · Together AI

OPERATE

Diagnose a failed agent run from the trace you already have, in the tool you already use

Braintrust ingests OpenTelemetry traces directly and adds a Debugger and Analyze-trace view, so instrumentation you shipped for your APM now feeds failure diagnosis instead of requiring a second SDK. Mastra ships a Langfuse-to-Mastra trace migration CLI plus delta polling on trace pagination, so switching observability backends no longer means abandoning the historical runs you compare against.

> Braintrust

Does **Opus 5** read better? 👍 👎

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.92 NOTES ↗ RANK

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

> xalgorix v4.6.92 ships a Kubernetes Helm chart, a deterministic CVE benchmark suite with real-world fixtures, OAST-based RCE attribution, and a new lane-based professional assessment workflow.

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

01 **Helm chart for Kubernetes deployment** NEW 90

xalgorix now ships a `deploy/chart` Helm chart for deploying on Kubernetes, with optional HPA, Ingress, and Gateway API support, `env.raw`, `env.config`, and `env.secret` passthrough, `auth` and `existingSecret` for credentials, a `/data` PVC, and hardened defaults including `serviceAccount.automount: false`.

Deploy xalgorix on Kubernetes with an existing secret for auth credentials and a persistent data volume.

```
$ helm install xalgorix ./deploy/chart \
  --set auth.existingSecret=xalgorix-creds \
  --set serviceAccount.automount=false
```

Configure environment passthrough — mixing plain config values and a secret reference — using the Helm chart's `env.raw`, `env.config`, and `env.secret` keys.

yaml

```yaml
env:
  raw:
    LOG_LEVEL: info
  config:
    APP_ENV: production
  secret:
    API_KEY: xalgorix-api-secret
auth:
  existingSecret: xalgorix-creds
serviceAccount:
  automount: false
```

— Runnable helm command plus full config key list v4.6.92

02 **Deterministic CVE benchmark suite** NEW 65

Adds `verify_path_traversal`, `verify_timing`, `discover_client_routes`, and `verify_path_template_xss` deterministic verification primitives for black-box CVE benchmarking, along with a real-world CVE benchmark suite using digest-pinned fixtures for Grafana 8.2.2/8.2.7 (CVE-2021-43798, CVE-2021-41174) and Metabase 0.46.6/0.46.6.4 (CVE-2023-38646), loopback-only ingress, and fixed-version control targets.

— Names functions and CVEs but gives no usage steps v4.6.92

03 **OAST execution attribution for RCE proof** NEW 60

Adds OAST execution attribution for os-command, runtime-api (including `java.net.URL` compound), and template-execution callbacks, with an explicit rule that fetch-only callbacks cannot prove RCE.

— Explains rule and callback types, no example given v4.6.92

THINNER COVERAGE BELOW

04 **Professional assessment workflow with ledger tracking** NEW 40

Adds a professional assessment workflow with lane delegation, plan/ledger-based completion tracking, and fail-closed RCE reporting with in-place upgrades.

— Describes workflow concept only, no named surfaces v4.6.92

v4.6.92 NOTES ↗ LATEST STABLE

xalgorix v4.6.92 adds a deterministic CVE benchmark suite and a Helm chart for Kubernetes deployment.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.92
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.92
```

└──▷ TRY IT

Deploy xalgorix on Kubernetes with an existing secret for auth credentials and a persistent data volume.

```
$ helm install xalgorix ./deploy/chart \
  --set auth.existingSecret=xalgorix-creds \
  --set serviceAccount.automount=false
```

Configure environment passthrough — mixing plain config values and a secret reference — using the Helm chart's `env.raw`, `env.config`, and `env.secret` keys.

deploy/chart/values.yaml

```
env:
  raw:
    LOG_LEVEL: info
  config:
    APP_ENV: production
  secret:
    API_KEY: xalgorix-api-secret
auth:
  existingSecret: xalgorix-creds
serviceAccount:
  automount: false
```

Adds `deploy/chart` Helm chart for Kubernetes deployment with optional HPA, Ingress, and Gateway API support, `env.raw|config|secret` passthrough, `auth` + `existingSecret`, `/data` PVC, and hardened defaults including `serviceAccount.automount: false`.

Adds `verify_path_traversal`, `verify_timing`, `discover_client_routes`, and `verify_path_template_xss` deterministic verification primitives for black-box CVE benchmarking.

Adds OAST execution attribution for os-command, runtime-api (including `java.net.URL` compound), and template-execution callbacks, with an explicit rule that fetch-only callbacks cannot prove RCE.

Adds a real-world CVE benchmark suite with digest-pinned fixtures for Grafana 8.2.2/8.2.7 (CVE-2021-43798, CVE-2021-41174) and Metabase 0.46.6/0.46.6.4 (CVE-2023-38646), loopback-only ingress, and fixed-version control targets.

Adds a professional assessment workflow with lane delegation, plan/ledger-based completion, and fail-closed RCE reporting with in-place upgrades.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

Replit Agent

Feedback ⓘ changelog · Sep 23, 2026 CHANGELOG ↗

RANK

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

> Replit Agent adds beta custom API connectors for Pro and Enterprise teams and expands audit logging with 65+ new event types for compliance and investigations.

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

01 **Expanded audit log event coverage** IMPROVED 62

Audit logs gain 65+ additional event types covering projects, workspaces, deployments, account security, SSO and SCIM, connectors, secrets, and Agent activity, accessible via the audit log portal and configured SIEM destinations.

— Lists categories and access points but no specific event names.

changelog-20260923-9a35757e

THINNER COVERAGE BELOW

02 **Custom API connectors (beta)** NEW 58

Pro and Enterprise teams can let Agent connect to external APIs by supplying an endpoint, authentication, and a usage description, configured from the Integrations page.

— Names config location and inputs but no API/flag names. changelog-20260923-9a35757e

changelog · Sep 23, 2026 CHANGELOG ↗ LATEST STABLE

Replit Agent gains custom API connectors and 65+ new audit log events for Enterprise compliance and investigations.

Adds custom connectors (beta) for Pro and Enterprise teams, letting Agent connect to external APIs by supplying an endpoint, authentication, and a usage description — configured from the Integrations page.

Expands audit logs with 65+ additional event types covering projects, workspaces, deployments, account security, SSO and SCIM, connectors, secrets, and Agent activity, accessible via the audit log portal and configured SIEM destinations.

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.158.0-alpha.2 NOTES ↗

CODE ↗ RANK

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

> Codex CLI's biggest changes this window are the rollout of GPT-6 Sol and Luna models across the catalog and Bedrock, a new optional fullscreen `/tui` interface, default-on voice conversations, a `/usage` analytics dashboard, and a `--no-daemon` bypass flag, alongside proxy, MCP, and image-generation refinements and two breaking removals.

└──▷ **WHAT SHIPPED · 16 FEATURES** most completely described first ? what's the number?

01 **Transparent/opaque background control for image generation**

IMPROVED 80

Adds a `transparent_background` boolean argument to `image_gen.imagegen` for both generation and editing; `true` maps to `transparent` and `false` or omission maps to `opaque`, replacing the previous `auto` background setting.

— Names exact argument and value mapping, no example given rust-v0.158.0-alpha.2

02 **--no-daemon flag to bypass background server** NEW 80

Adds `--no-daemon` flag to bypass the local background server, e.g. `codex --no-daemon` for ephemeral or isolated CI environments, or update it via the `/daemon` command.

Run Codex without starting or connecting to the local background daemon — useful for ephemeral or isolated CI environments.

```
$ codex --no-daemon
```

— Exact flag with runnable example command rust-v0.156.0

03 **/tui fullscreen interface with transcript search** NEW 70

Adds `/tui` command to launch an optional fullscreen UI with transcript search, mouse selection, and right-click copying.

— Names exact command and its capabilities rust-v0.156.0

04 **GPT-6 Sol and Luna model catalog rollout** NEW 65

Adds `gpt-6-sol` and `gpt-6-luna` to the model catalog, including Amazon Bedrock catalogs, with migrations from `gpt-5.5`, `gpt-5.6-sol`, `gpt-5.6-terra`, and `gpt-5.6-luna`; the rate-limit switch prompt now recommends `gpt-6-luna` instead of previous models.

— Names all model ids and migration paths, no command shown `rust-v0.155.0-alpha.16.3` `rust-v0.156.1` `rust-v0.157.0-alpha.10`

---

05 **Default-on voice conversations with F8 toggle** NEW

65

Adds `/voice settings` picker to choose a voice for future conversations; voice conversations are now enabled by default and can be toggled via F8. Native voice runtimes are now bundled for Linux and Windows releases.

— Names command and toggle key, no runnable command shown `rust-v0.156.0`

---

06 **/usage analytics dashboard** NEW

65

Adds `/usage` analytics dashboard to explore account usage, token totals, and plugin and skill activity.

— Names exact command with clear scope of data shown `rust-v0.156.0`

THINNER COVERAGE BELOW

---

07 **/import command in remote and local daemon sessions** NEW

55

Enables `/import` in both remote and local daemon sessions.

— Names exact command, no usage example given `rust-v0.157.0-alpha.10`

---

08 **SQLite database failure detection in codex doctor** NEW

50

Identifies failed SQLite databases in `codex doctor` output.

— Names diagnostic command, minimal mechanism detail `rust-v0.157.0-alpha.10`

---

09 **Worktree session creation in agent command center** NEW

50

Adds worktree session creation to the agent command center, with worktree support now enabled by default.

— Names feature area but no exact command or path `rust-v0.156.0`

---

10 **Faster agent startup via parallelized instruction refresh**

IMPROVED

45

Parallelizes `AGENTS.md` and thread-instruction refresh concurrently with capability discovery, MCP setup, and tool construction, reducing agent startup latency.

– Explains mechanism but no user-facing action `rust-v0.158.0-alpha.2`

---

11 **MCP OAuth endpoints restricted to HTTP(S)** IMPROVED

45

Restricts MCP OAuth authorization endpoints to HTTP(S).

– Security policy change stated without further mechanism `rust-v0.157.0-alpha.10`

---

12 **Personality settings removed from TUI** BREAKING

45

The `friendly` and `pragmatic` personality settings no longer select response styles — they are deprecated and personality selection has been removed from the TUI.

– Names removed settings, gives no migration path `rust-v0.156.0`

---

13 **HTTP origins exposed in MCP server status** NEW

40

Exposes HTTP origins in MCP server status.

– Bare one-line addition with no further detail `rust-v0.157.0-alpha.10`

---

14 **Fork conversations locked by another app in TUI** NEW

40

Allows forking conversations locked by another app in the TUI.

– Describes capability but no exact UI path `rust-v0.157.0-alpha.10`

---

15 **Explicit gateway login control and auth status** NEW

35

Adds explicit gateway login control and authentication status.

– Thin one-line description, no named command `rust-v0.157.0-alpha.10`

---

16 **thread/rollback API removed** BREAKING

30

The `thread/rollback` API has been removed.

– Bare statement of removal with no replacement noted `rust-v0.156.0`

└──▷ **BREAKING ON UPGRADE**

! The `friendly` and `pragmatic` personality settings no longer select response styles — they are deprecated and personality selection has been removed from the TUI.

! The `thread/rollback` API has been removed.

> `rust-v0.158.0-alpha.2` NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI adds explicit transparent/opaque background control to image generation and an option to disable multi-agent v2 direct messaging.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rust-v0.158.0-alpha.2
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.158.0-alpha.2
```

Adds `transparent_background` boolean argument to `image_gen.imagegen` for generation and editing, mapping `true` to `transparent` and `false` or omission to `opaque`, replacing the previous `auto` background setting.

Parallelizes `AGENTS.md` and thread-instruction refresh concurrently with capability discovery, MCP setup, and tool construction, reducing agent startup latency.

`rust-v0.155.0-alpha.16.3` NOTES ↗ CODE ↗ PRE-RELEASE

Adds system proxy fallback for login/startup and expands model catalog with `gpt-6-sol` and `gpt-6-luna`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rust-v0.155.0-alpha.16.3
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.155.0-alpha.16.3
```

Adds `gpt-6-sol` and `gpt-6-luna` to the model catalog, with migrations from `gpt-5.5`, `gpt-5.6-sol`, `gpt-5.6-terra`, and `gpt-5.6-luna`.

Enables `features.system_proxy_fallback` by default so login and enterprise configuration bootstrap succeed when endpoints are reachable only through the system proxy, before cloud configuration can enable `respect_system_proxy`.

`rust-v0.156.1` NOTES ↗ CODE ↗ LATEST STABLE

Codex CLI adds GPT-6 Sol and GPT-6 Luna to the model picker, with Luna recommended at rate-limit.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rust-v0.156.1
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.156.1
```

Rate-limit switch prompt now recommends `gpt-6-luna` instead of previous models.

rust-v0.157.0-alpha.10 NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI adds GPT-6 Sol and Luna models, Guardian extra-policy config, MITM CA support, and fullscreen transcript on by default.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rust-v0.157.0-alpha.10
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.157.0-alpha.10
```

Supports caller-provided MITM certificate authorities in the network proxy.

Adds GPT-6 Sol and Luna to the model catalog, including Amazon Bedrock catalogs.

Enables `/import` in both remote and local daemon sessions.

Identifies failed SQLite databases in `codex doctor` output.

Exposes HTTP origins in MCP server status.

+6 more

rust-v0.156.0 NOTES ↗ CODE ↗ STABLE

Codex CLI gains an optional fullscreen TUI, default-on voice with F8 toggle, a `/usage` dashboard, and `--no-daemon` bypass.

└──▷ GET THIS VERSION ▾

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

└──▷ TRY IT

Run Codex without starting or connecting to the local background daemon — useful for ephemeral or isolated CI environments.

```
$ codex --no-daemon
```

Adds `--no-daemon` flag to bypass the local background server, or update it via the `/daemon` command.

Adds `/tui` command to launch an optional fullscreen UI with transcript search, mouse selection, and right-click copying.

Adds `/voice settings` picker to choose a voice for future conversations, with voice conversations now enabled by default and toggled via F8.

Adds `/usage` analytics dashboard to explore account usage, token totals, and plugin and skill activity.

Adds worktree session creation to the agent command center, with worktree support now enabled by default.

+2 more

└─▷ **BREAKING ON UPGRADE**

! The `friendly` and `pragmatic` personality settings no longer select response styles — they are deprecated and personality selection has been removed from the TUI.

! The `thread/rollback` API has been removed.

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.66 NOTES ↗

RANK

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

> Graphify v0.9.66 broadens language coverage with five new extractors — VB.NET, R, Solidity, Erlang, and COBOL — for building knowledge graphs from code, alongside improved resolution for Go and Rust.

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

01 **New language extractors: VB.NET, R, Solidity, Erlang, COBOL**

NEW 89

v0.9.66 adds five new language extractors. The VB.NET extractor supports `.vb` files with case-insensitive types/methods and Inherits / Implements / Handles relationships. The R extractor supports `.r` / `.R` files with assignment-form function definitions, `library` / `source` imports, and S4/R6 classes, shipped via the `r` extra (`pip install 'graphifyy[r]'`). The Solidity extractor supports `.sol` files covering contracts, interfaces, libraries, `is` inheritance, imports, and modifiers, shipped via the `solidity` extra (`pip install 'graphifyy[solidity]'`). The Erlang extractor supports `.erl` / `.hrl` / `.escript` files covering modules, functions by arity, behaviours, and local and remote foo:bar() calls, shipped via the `erlang` extra (`pip install 'graphifyy[erlang]'`), with no standalone PyPI wheel. A COBOL extractor was also added but no further specifics were given in this release.

Map a Solidity smart-contract repo — install the new `solidity` extra first, then build the graph.

```
$ pip install 'graphifyy[solidity]'
graphify .
```

Map an Erlang codebase, pulling in the `erlang` extra (no standalone PyPI wheel — must be installed this way).

```
$ pip install 'graphifyy[erlang]'
graphify .
```

Map an R project, pulling in the `r` extra to enable S4/R6 class and `library` / `source` edge extraction.

```
$ pip install 'graphifyy[r]'
graphify .
```

— Names extras, extensions, relationships for four extractors; COBOL undetailed. v0.9.66

THINNER COVERAGE BELOW

02 **Improved Go and Rust resolution** IMPROVED 13

v0.9.66 includes improved resolution for Go and Rust code, but the release notes give no further mechanism or scope beyond naming the two languages.

— Only named in the summary line, no detail given. v0.9.66

v0.9.66 NOTES ↗ LATEST STABLE

Graphify v0.9.66 adds five new language extractors — COBOL, VB.NET, R, Solidity, and Erlang — plus improved Go and Rust resolution.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.66 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.66
```

└──▷ TRY IT

Map a Solidity smart-contract repo — install the new `solidity` extra first, then build the graph.

```
$ pip install 'graphifyy[solidity]'
graphify .
```

Map an Erlang codebase, pulling in the `erlang` extra (no standalone PyPI wheel — must be installed this way).

```
$ pip install 'graphifyy[erlang]'
graphify .
```

Map an R project, pulling in the `r` extra to enable S4/R6 class and `library` / `source` edge extraction.

```
$ pip install 'graphifyy[r]'
graphify .
```

Adds VB.NET extractor supporting `.vb` files with case-insensitive types/methods, `Inherits` / `Implements` / `Handles` relationships.

Adds R extractor supporting `.r` / `.R` files with assignment-form function defs, `library` / `source`, and S4/R6 classes; ships via the `r` extra.

Adds Solidity extractor supporting `.sol` files covering contracts, interfaces, libraries, `is` inheritance, imports, and modifiers; ships via the `solidity` extra.

Adds Erlang extractor supporting `.erl` / `.hrl` / `.escript` files covering modules, functions by arity, behaviours, and local + remote foo:bar() calls; ships via the `erlang` extra.

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ v4.1.20 NOTES ↗ CODE ↗ RANK

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

> Cline added remote SSH execution for its Desktop agent and shipped a release that parallelizes sub-agent tool calls, expands its model catalog to over 6,000 models, and changes default model resolution for many providers.

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

01 **Parallel sub-agent tool calls** IMPROVED 50

Sub-agents spawned in the same step now run their tool calls in parallel rather than sequentially. Tools that must execute in order still do so, and the parent agent waits for all sub-agent results before taking its next turn.

– Clear mechanism explained but nothing for a reader to invoke v4.1.20

02 **Remote execution over SSH in Cline Desktop** NEW 45

Cline Desktop can now run the Cline agent, its tools, and code on a remote dev box, build server, or cloud VM over SSH, while the desktop app itself stays local.

– Describes the capability but no setup steps or config named product docs

03 **Expanded model catalog and output budgets** IMPROVED 45

The model catalog now covers 6,237 models across 209 providers, and output budgets were raised for large-context models.

– Concrete numbers given but no mechanism or usage path v4.1.20

└──▷ **BREAKING ON UPGRADE**

! The resolved default model changes for 36 providers that do not pin a model — most landing on DeepSeek V4.1 Flash, GLM 5.3 Flash, or MiMo V2.6 Flash. If you use one of those providers without pinning a model, your tasks will run against a different model after upgrading.

v4.1.20 NOTES ↗ CODE ↗ LATEST STABLE

Cline v4.1.20 parallelizes sub-agent tool calls, raises output budgets for large-context models, and expands the model catalog to 6,237 models across 209 providers.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v4.1.20 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout v4.1.20
```

Sub-agents spawned in the same step now run their tool calls in parallel rather than sequentially, while tools that must run in order still do and the parent waits for all results before its next turn.

└─▷ **BREAKING ON UPGRADE**

! The resolved default model changes for 36 providers that do not pin a model — most landing on DeepSeek V4.1 Flash, GLM 5.3 Flash, or MiMo V2.6 Flash. If you use one of those providers without pinning a model, your tasks will run against a different model after upgrading.

WAS THIS USEFUL? 👍 👎

Superset

Feedback ⓘ desktop-v1.30.2 NOTES ↗ RANK

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

> Superset desktop now lets users browse and clone GitHub repositories directly within the app.

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

01 **GitHub repository picker in desktop app** NEW 40

Adds the ability to pick GitHub repositories to clone from within the desktop app, browsing or searching your repositories and selecting one to clone locally without leaving the app.

Clone a GitHub repository without leaving the desktop app — useful when starting work on a new project.

> 📍 In the desktop app, open the repository picker and browse or search your GitHub repositories, then select one to clone locally.

— Clear UI flow described but no config or command names given desktop-v1.30.2

desktop-v1.30.2 NOTES ↗ LATEST STABLE

Superset desktop v1.30.2 lets you browse and clone GitHub repositories directly from the app.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch desktop-v1.30.2 https://github.com/superset-sh/superset.git
# already have the repo? check out this version:
$ git checkout desktop-v1.30.2
```

└──▷ **HOW TO FIND IT**

Clone a GitHub repository without leaving the desktop app — useful when starting work on a new project.

> 📍 In the desktop app, open the repository picker and browse or search your GitHub repositories, then select one to clone locally.

Adds the ability to pick GitHub repositories to clone from within the desktop app.

WAS THIS USEFUL?  

GitHub Copilot CLI

Feedback ⓘ v1.0.88 NOTES ↗ RANK

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

> Copilot CLI v1.0.88 extends enterprise managed-settings enforcement to ACP, AHP-host, and server-mode sessions, adds a `/fork` command for branching mid-turn, and ships a cluster of smaller improvements to indexed search, custom skills, session management, and MCP/plugin visibility.

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

01 Per-agent reasoning-effort application IMPROVED 85

A custom agent's `reasoning-effort` now applies when the agent is selected rather than only its model; a level unsupported by the selected model is reported and left unapplied, with an explicit `--reasoning-effort` flag still taking precedence.

Override reasoning effort at launch while letting the agent's own `reasoning-effort` setting serve as the baseline for all other sessions.

```
$ copilot --reasoning-effort high
```

— Names field, flag, precedence rule, and includes runnable example. v1.0.88

02 Session-dismissal confirmation in Sessions tab NEW 80

Adds session-dismissal confirmation in the Sessions tab: press `x` then `x` again — footer text indicates whether the local session will be permanently deleted or only closed on a server-backed session.

— Exact keybinding and UI behaviour spelled out. v1.0.88

03 Managed-settings enforcement for ACP, AHP-host, server sessions

IMPROVED 75

Adds `--acp`, `--ahp-host`, and `--server` session coverage for enterprise managed settings, so MCP, permission, and plugin policies now apply to ACP-mode, AHP-host, and published server sessions that previously ran with no policy enforcement.

— Names exact session flags and before/after but no user command. v1.0.88

04 /fork command during active turns NEW 70

Adds the `/fork` slash command, usable during an active turn to branch work into a separate thread without waiting for the current turn to finish.

Branch an in-progress investigation into a separate thread without waiting for the current agent turn to complete.

```
$ /fork
```

— Exact command and usage example given, mechanism is brief. v1.0.88

05 **Allow-all path grant visibility and reset** NEW 70

Exact path grants from `/allow-all` are now visible in `/list-dirs` and can be cleared with `/reset-allowed-tools`.

— Three exact commands named, simple mechanism. v1.0.88

06 **Glob-filtered indexed search with --files listing** IMPROVED 65

Adds glob filtering and `--files` listing support to indexed search, with accurate ripgrep fallback behavior.

— Names the flag and fallback mechanism, no example command. v1.0.88

THINNER COVERAGE BELOW

07 **Plugin-dir agents available in server-mode sessions** NEW 55

Adds `--plugin-dir` `-mounted` agents to server-mode sessions, making custom plugin agents available when Copilot runs as a server.

— Names the flag but describes only what it enables, not mechanism. v1.0.88

08 **Ask_user prompt newline and submit keys** IMPROVED 55

Pressing Enter in freeform `ask_user` prompts adds a new line; submit with `Ctrl+Enter` or `Ctrl+S`.

— Exact keys named for a small interaction change. v1.0.88

09 **OSC 777 terminal notifications** NEW 45

Adds OSC 777 terminal notifications for direct Ghostty and WezTerm sessions.

— Names the terminals and protocol but no usage detail. v1.0.88

10 **Namespaced custom skills and ignored directories** NEW 45

Supports namespaced custom skills and ignored skill directories during skill discovery.

— States the capability without configuration detail. v1.0.88

11 **MCP and plugin view clarity improvements** IMPROVED 45

MCP and plugin views now show server display names and plugin descriptions for clearer status, and deferred MCP tools are now listed under their sanitized or shortened registered name so tool search can find them.

— Describes two thin UI/search fixes without concrete surfaces. v1.0.88

v1.0.88 NOTES ↗ LATEST STABLE

Copilot CLI v1.0.88 adds OSC 777 notifications, `/fork` during active turns, glob-filtered indexed search, namespaced custom skills, and broader enterprise managed-settings coverage.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.0.88 https://github.com/github/copilot-cli.git
# already have the repo? check out this version:
$ git checkout v1.0.88
```

└──▷ TRY IT

Branch an in-progress investigation into a separate thread without waiting for the current agent turn to complete.

```
$ /fork
```

Override reasoning effort at launch while letting the agent's own `reasoning-effort` setting serve as the baseline for all other sessions.

```
$ copilot --reasoning-effort high
```

Adds `--acp`, `--ahp-host`, and `--server` session coverage for enterprise managed settings, so MCP, permission, and plugin policies now apply to ACP-mode, AHP-host, and published server sessions that previously ran with no policy enforcement.

Adds `--plugin-dir` `-mounted` agents to server-mode sessions, making custom plugin agents available when Copilot runs as a server.

Adds `/fork` slash command usable during active turns to branch work without waiting for the current turn to finish.

> Adds OSC 777 terminal notifications for direct Ghostty and WezTerm sessions.
>
> Adds glob filtering and `--files` listing support to indexed search, with accurate ripgrep fallback behavior.
>
> +7 more

WAS THIS USEFUL? 👍 👎

Earendil Works Pi

Feedback ⓘ v0.87.1 NOTES ↗ RANK

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

> Pi shipped a broad CLI and extensibility overhaul this window: scriptable `--print` / `--mode json` / `--mode rpc` output with a typed RpcClient, a full package system spanning extensions/skills/prompts/themes, new Skills and Prompt Template systems, four ways to run Pi in isolation (Docker, Docker Sandbox, OpenShell, Gondolin), llama.cpp router support for local GGUF models, and new frontier model support (Claude Opus 5.5, GPT-6 Sol/Luna, Grok 4.7 default).

└──▷ **WHAT SHIPPED · 22 FEATURES** most completely described first ? what's the number?

01 Package management: install, sources, manifest, gallery NEW

93

`pi install npm:<pkg>@<ver>`, `pi install git:<repo>@<ref>`, and `pi install <./local-path>` install packages from npm, git repos, or local directories; `pi list` shows configured packages and `pi remove <source>` (also `pi uninstall`) removes one; `pi -e npm:<pkg>` tries a package for a single invocation without persisting it; `--local` / `-l` writes the package declaration to project-local settings instead of personal settings; `pi update`, with `--extensions`, `--models`, `--all`, and `--force` targets, updates selectively or forcibly; `pi config` (with `--local` and Tab to switch scope) enables or disables discovered package resources interactively. A `pi` manifest key in `package.json` declares `extensions`, `skills`, `prompts`, and `themes` fields (glob patterns supported), and object-form package entries in settings can filter which resources load per type. A `pi` keyword in `package.json` makes an npm package discoverable in the Pi package gallery, with optional `pi.image` and `pi.video` preview fields. Versioned npm specs and git tags/commits are pinned, and git sources reconcile to the configured ref without moving it on update; local file paths load a single extension while local directory paths follow full package discovery (TypeScript/JavaScript extensions, skill directories, Markdown prompts, JSON themes).

Install a pinned npm package of Pi extensions/skills for the current project so teammates get the same toolset after granting project trust.

```
$ pi install --local npm:@example/pi-tools@1.0.0
```

Try a community package for one session before committing — useful for evaluating a third-party skill set without touching settings.

```
$ pi -e npm:@example/pi-tools
```

— Every command, flag, and manifest key named with runnable examples `product docs`

02 **llama.cpp router integration for local GGUF models** NEW

93

`/llama` interactively loads, unloads, and downloads GGUF models from a running llama-server router, including Hugging Face search with `owner/repository[:quant]` syntax; `/login llama.cpp` configures the router URL and optional API key, or `LLAMA_BASE_URL` / `LLAMA_API_KEY` env vars can be used instead for scripted setup. Loaded and sleeping models appear in `/model`, and sleeping models wake automatically when selected. Hugging Face tokens are discovered via `HF_TOKEN`, `$HF_TOKEN_PATH`, `$HF_HOME/token`, `$XDG_CACHE_HOME/huggingface/token`, or `~/.cache/huggingface/token`. Integration uses llama-server router mode flags `--models-dir`, `--no-models-autoload`, `--jinja`, `-ngl`, and `-c` to discover, GPU-offload, and context-tune multiple GGUF models on demand.

Start the llama-server in router mode to serve multiple local GGUF models with GPU offload and a 32 K context window, then point Pi at it.

```
$ llama-server \
  --models-dir ~/models \
  --no-models-autoload \
  --jinja \
  --host 127.0.0.1 \
  --port 8080 \
  -ngl 999 \
  -c 32768
```

Configure Pi to talk to the local llama-server router without an interactive `/login` prompt — useful in CI or dotfile setup.

```
$ export LLAMA_BASE_URL=http://127.0.0.1:8080
export LLAMA_API_KEY=optional-secret
pi
```

Verify the router is reachable and confirm which models are currently known before running `/llama` inside Pi.

```
$ curl http://127.0.0.1:8080/health
curl http://127.0.0.1:8080/models
```

— Commands, env vars, and flags all named with runnable examples `product docs`

03 **Scriptable print, JSON, and RPC output modes** NEW 91

`--print` / `-p` runs a prompt once and writes the final assistant text to stdout then exits, with automatic print-mode selection when stdin/stdout is non-TTY, errors on stderr, and nonzero exit on `error` / `aborted` stop reasons. `--mode json` streams a session header plus agent/session events as JSONL to stdout, with `message_update` records carrying deltas and `agent_settled` marking end of automatic work. `--mode rpc` keeps Pi running as a long-lived process reading JSON commands from stdin and streaming responses, enabling bidirectional control of model changes, session management, shell commands, and extension UI requests. `--export <input> [output]` exports a session file to HTML and exits. `RpcClient` from `@earendil-works/pi-coding-agent` gives typed Node.js/TypeScript integration, starting a Pi RPC child process, correlating requests, and exposing typed methods including promptAndWait() and waitForIdle().

Pipe a git diff into Pi to get a one-shot code review written to stdout — useful in CI or pre-push hooks.

```
$ git diff | pi --print "Review this change"
```

Pipe a repo diff into Pi and capture only the final assistant text in a CI script — no structured parsing needed.

```
$ git diff HEAD~1 | pi --print "Summarize these changes" >
summary.txt
```

Capture every agent and tool event from a review run for downstream processing or replay.

```
$ pi --mode json "Review this repository" > events.jsonl
```

— Full mechanism, formats, and runnable examples given `product docs`

04 **Skills system for lazily-loaded instruction bundles** NEW 91

Skills are directory-based bundles of instructions, scripts, and reference files that Pi advertises by name/description and loads into context only when a task matches. Setting `disable-model-invocation: true` in frontmatter restricts a skill to explicit `/skill:<name>` invocation, bypassing automatic model selection; the `enableSkillCommands` setting controls whether skill commands appear in interactive command discovery (manual entry still works). `/skill:<name>` force-loads a skill, appending any following arguments to its instructions, and `/reload` hot-reloads skill changes without restarting. Pi discovers project `skill.md` directories recursively from the working directory up through ancestors, stopping at the repository root, and skills can be distributed as Pi packages via npm or git.

Force-load a specific skill and pass a task in one shot — useful in CI or scripted workflows where you cannot rely on automatic model routing.

```
$ /skill:pdf-tools extract report.pdf
```

Lock a sensitive or specialized skill so it only activates when explicitly invoked, never auto-selected by the model.

```yaml
name: pdf-tools
description: Extract text and tables from PDF files. Use when
reading, converting, or inspecting PDFs.
disable-model-invocation: true
```

Hot-reload an edited skill mid-session without restarting Pi, so changes take effect immediately during development.

```
$ /reload
```

— Frontmatter key, commands, and discovery rules named with examples product docs

05 **Custom theming system with hot-reload** NEW 90

A `theme` setting (e.g. `"theme": "dark"`) persists the active theme, with a `"theme": "light/dark"` automatic mode that switches when the terminal reports an appearance change. Custom JSON theme files live at `<agent-dir>/themes/<name>.json` (default `~/.pi/agent/themes/`) or project-scoped in `.pi/themes/` (loaded only after project trust), hot-reloaded on `/reload`. The theme JSON schema has a `colors` block for reusable variables (with chained references), a `roles` block mapping colors to terminal UI roles, and an optional `export` block (`export.pageBg`, `export.cardBg`, `export.infoBg`) for HTML export backgrounds; colors accept six-digit RGB hex (`#00aaff`), 256-color ANSI index, a

variable reference, or the terminal default foreground/background. Themes can be distributed via Pi packages or loaded from arbitrary paths, and duplicate theme names are reported as resource collisions.

Try a dark theme for one session without overwriting your saved preference — useful when demoing or testing a new palette.

```
$ pi --use-theme dark
```

Enable automatic light/dark switching so Pi follows your terminal's appearance change events.

```json
{
  "theme": "light/dark"
}
```

Create a custom theme that overrides syntax highlighting and HTML export backgrounds, then hot-reload it without restarting Pi.

```json
{
  "name": "my-theme",
  "colors": {
    "accent": "#00aaff",
    "accentDim": "accent"
  },
  "roles": {
    "syntax.keyword": "accent",
    "syntax.string": "#ffaa00"
  },
  "export": {
    "pageBg": "#1a1a2e",
    "cardBg": "#16213e",
    "infoBg": "#0f3460"
  }
}
```

— Full schema, file paths, and examples given `product docs`

06 **Session lifecycle flags and slash commands** NEW 83

`--session <path|id>` opens a session by file path, exact ID, or partial ID with a cross-project fork offer; `--session-id <id>` opens or creates an exact project session by ID; `--fork <path|id>` forks an existing session into a new project session

before interactive mode starts; `--session-dir <dir>` overrides session storage/lookup, taking precedence over `PI_CODING_AGENT_SESSION_DIR` and the `sessionDir` setting; `--no-session` uses an in-memory session that is never persisted; `--continue` / `-c` resumes the most recent session for the current project; `--resume` / `-r` opens the session selector. In-session, `/name` assigns a recognizable name to the current session and `/export` writes it as HTML or JSONL, with an upload option producing a viewer link via a Radius artifact or a private GitHub gist.

Resume a previous working session for the current directory without losing context — useful at the start of a new terminal session.

```
$ pi --continue
```

Store all sessions in a shared team directory so multiple users can access session history.

```
$ pi --session-dir /shared/team/pi-sessions
```

Branch from an existing session to explore an alternative approach without disturbing the original conversation.

```
$ pi --fork <session-path>
```

— Every flag and command named with runnable examples `product docs`

07 **Prompt templates as reusable slash commands** NEW 83

Markdown files in `~/.pi/agent/prompts/` become reusable slash commands, with the filename becoming the command name (e.g. `review.md` → `/review`). An `argument-hint` front-matter field surfaces a hint in command completion, and shell-like quoting lets `/review "API compatibility"` pass one multi-word argument. Templates load from personal configuration (`~/.pi/agent/prompts/`), project configuration (after trust), explicit paths, and Pi packages (manifests support explicit paths and globs for nested discovery); `/reload` picks up new or changed templates without restarting.

Create a reusable code-review slash command that focuses on a caller-supplied topic, defaulting to correctness and security when no argument is given.

```
$ mkdir -p ~/.pi/agent/prompts
cat > ~/.pi/agent/prompts/review.md <<'EOF'
---
argument-hint: "[focus]"
---
Review the staged changes. Focus on ${1:-correctness, security,
and error handling}.
EOF
```

Invoke the template from the Pi editor with a multi-word argument to run a targeted review without retyping the full prompt.

```
$ /review "API compatibility"
```

After adding or editing a template in a running Pi session, reload templates without restarting.

```
$ /reload
```

— File location, syntax, and reload command all runnable product docs

08 **Tool selection flags and built-in tool set** NEW 80

`--tools <list>` / `-t` replaces the default tool selection with a comma-separated allowlist of built-in, extension, or custom tools; `--exclude-tools <list>` / `-xt` disables specific tools after other selection; `--no-builtin-tools` / `-nbt` disables default built-in tools while keeping extension/custom tools; `--no-tools` / `-nt` disables all tools at start. Built-in tools exposed for selection are `read`, `bash`, `edit`, `write`, `grep`, `find`, `ls`, and `powershell` (Windows).

Run Pi non-interactively with a restricted tool set so it can only read and search files — no writes or shell execution — during an automated audit.

```
$ pi --tools read,grep,find,ls --print "Review this project"
```

— Names every flag and tool with a runnable example product docs

09 **Credential inspection commands** NEW 78

`pi auth check`, `pi auth print-api-key`, and `pi auth print-bearer-token` inspect and print resolved credentials, with `--provider`, `--model`, `--json`, `--credentials`, `--no-refresh`, and `--min-expiry <duration>` options.

Verify that credentials for a provider are correctly resolved before running a scan, outputting structured JSON for parsing in a pipeline.

```
$ pi auth check --provider openai --json
```

— Names every subcommand and flag with a runnable example product docs

10 **OpenShell sandbox with policies and file transfer** NEW

78

Pi can run inside an OpenShell sandbox governed by filesystem, process, network, credential, and inference policies, including optional remote sandboxes. `openshell sandbox upload` and `openshell sandbox download` transfer files to and from a remote OpenShell sandbox.

Transfer your local working folder into a remote OpenShell sandbox before a Pi session so Pi can access your project files without a bind mount.

```
$ openshell sandbox upload pi-sandbox ./working-folder /workspace
```

Retrieve results from a remote OpenShell sandbox after Pi has finished processing, pulling the output back to your host machine.

```
$ openshell sandbox download pi-sandbox /workspace/working-folder
./working-folder-out
```

— Policy types and transfer commands named with examples product docs

11 **Model selection and catalog flags** NEW

75

`--model <pattern>` selects via fuzzy ID/name matching, an optional `provider/id` prefix, and an optional `:<thinking>` suffix; `--thinking <level>` sets thinking depth (`off` , `minimal` , `low` , `medium` , `high` , `xhigh` , `max`); `--models <patterns>` sets a comma-separated scope for startup and cycling (exact IDs, fuzzy matches, case-insensitive globs, `:<thinking>` suffixes); `--list-models [search]` lists available models with optional fuzzy filtering; `--provider <name>` restricts `--model` lookup to one provider; `--api-key <key>` gives a non-persistent per-invocation override; `--offline` (equivalent to `PI_OFFLINE=1`) disables automatic network activity including model catalog refreshes.

— Names every flag and value, no worked example product docs

12 **Extension, skill, prompt, and theme loading flags** NEW

75

`--extension <path>` / `-e` (repeatable) loads an extension file or directory and `--no-extensions` / `-ne` disables discovered extensions; `--skill <path>` (repeatable) and `--no-skills` / `-ns` load or disable skill files; `--prompt-template <path>` (repeatable) and `--no-prompt-templates` / `-np` load or disable prompt-template files; `--theme <path>` (repeatable), `--use-theme <name[/name]>`, and `--no-themes` load, select, or disable themes; `--no-context-files` / `-nc` disables `AGENTS.md` and `CLAUDE.md` discovery; `--system-prompt <text|path>` replaces the default system prompt and `--append-system-prompt <text|path>` (repeatable) appends to it.

— Every flag named, no worked example given `product docs`

13 **Bash executable and alias-expansion config** NEW 75

`shellPath` specifies the Bash executable Pi uses, overriding the default resolution order (`/bin/bash` , `bash` on PATH, then `sh`); `shellCommandPrefix` prepends shell setup — such as environment exports or alias loading — before every `bash` tool call and `!` / `!!` user command. Setting `shellCommandPrefix` to `shopt -s expand_aliases\nsource ~/.bash_aliases` enables Bash alias expansion.

Use a non-system Bash build (e.g. Homebrew or Cygwin) for all Pi shell commands.

```json
"shellPath": "~/.local/bin/bash"
```

Make shell aliases defined in `~/.bash_aliases` available to every command Pi runs, including `!` and `!!` shortcuts.

```json
"shellCommandPrefix": "shopt -s expand_aliases\nsource ~/.bash_aliases"
```

— Exact config keys and values given with examples `product docs`

14 **New model support: Claude Opus 5.5, GPT-6 Sol/Luna, Grok 4.7** NEW 63

Adds Claude Opus 5.5 via Anthropic with adaptive thinking and a 1M context window; adds GPT-6 Sol and GPT-6 Luna via OpenAI API keys and OpenAI Codex subscriptions; adds Claude Opus 5.5, GPT-6 Sol, and GPT-6 Luna through GitHub Copilot; new xAI sessions now default to Grok 4.7 instead of the previous default model.

— Names models, providers, and context window, no usage example `v0.87.1`

15 **Trust, TUI mode, and file-reference flags** NEW 60

`--approve` / `-a` and `--no-approve` / `-na` trust or ignore project-local configuration and resources for a single process; `--tui-mode <mode>` accepts `regular` or `fullscreen` to control terminal UI rendering; `@path` input syntax includes a text file or image in the first prompt, resolved from the current working directory.

— Flags and syntax named, no example shown product docs

16 **Gondolin local micro-VM extension** NEW 60

Gondolin is a local Linux micro-VM extension that keeps the Pi process and file-based provider credentials on the host while routing built-in tools and user commands into the VM; it requires Node.js 23.6 or newer and QEMU.

— Names requirements and mechanism, no setup command product docs

THINNER COVERAGE BELOW

17 **Manual and automatic context compaction** NEW 55

`/compact` manually compacts conversation history, with optional instructions to preserve specific topics or decisions. Pi also compacts automatically, adding a summary and retaining recent messages when the active context approaches the model's limit, without deleting original session entries.

— Explains trigger and behaviour, no numeric limit given product docs

18 **Source-fork rebranding via piConfig** NEW 50

Forks of the Pi source can rebrand the CLI by setting `piConfig.name` and `piConfig.configDir` in `package.json`, controlling the CLI banner, configuration paths, and derived environment variable names.

— Names exact config keys but gives no example product docs

19 **Bug-report bundling workflow** NEW 50

A bug-report workflow bundles environment and provider configuration (without credentials), recorded error diagnostics, and an optional session transcript or model-generated summary, for upload or export as a zip.

— Lists contents but no trigger command given product docs

20 **Run Pi inside a plain Docker container** NEW 50

Pi can run its complete process inside a plain Docker container, mounting a working folder and using a named `pi-agent-home` volume to persist container-local settings, credentials, and sessions between runs.

— Names the volume but gives no runnable command product docs

21 **Docker Sandbox with credential-proxying** NEW 48

Pi can run inside a Docker Sandbox (managed sandbox) whose proxy keeps the real provider credential on the host and substitutes it for outbound requests, preventing real credentials from being written into the sandbox.

— Explains proxy mechanism, no setup steps `product docs`

---

22 **Session branching as a tree** NEW 40

Sessions branch as a tree structure so returning to an earlier point does not erase the abandoned branch, and branch summaries can be attached when switching branches.

— Describes behaviour, no command shown `product docs`

v0.87.1 NOTES ↗ LATEST STABLE

Pi v0.87.1 adds Claude Opus 5.5, GPT-6 Sol/Luna, and switches xAI default to Grok 4.7

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.87.1 https://github.com/earendil-works/pi.git
# already have the repo? check out this version:
$ git checkout v0.87.1
```

Adds Claude Opus 5.5 support via Anthropic with adaptive thinking and a 1M context window.

Adds GPT-6 Sol and GPT-6 Luna support via OpenAI API keys and OpenAI Codex subscriptions.

Adds Claude Opus 5.5, GPT-6 Sol, and GPT-6 Luna support through GitHub Copilot.

New xAI sessions now default to Grok 4.7 instead of the previous default model.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.280 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.280 adds the Opus 5.5 model as the new default across most plans, an environment variable to raise MCP description length limits, richer hook telemetry, and small mouse-navigation improvements.

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

01 **MCP description length override** NEW 88

Adds the `CLAUDE_CODE_MAX_MCP_DESCRIPTION_LENGTH` environment variable to override the 2,048-character cap on MCP tool descriptions and server instructions for every MCP server in the session.

Allow verbose MCP servers whose tool descriptions exceed the 2,048-character default to load fully, without truncation, in your Claude Code session.

```
$ CLAUDE_CODE_MAX_MCP_DESCRIPTION_LENGTH=8192 claude
```

— Named env var with default limit and a runnable command v2.1.280

02 **Claude Opus 5.5 model added** NEW 68

Adds Claude Opus 5.5 (`claude-opus-5-5`) as the new default Opus model, with a 1M context window and $4/$20 per Mtok input/output pricing ($0.20/Mtok for cache reads).

— Names model id, context size, and pricing but no usage steps v2.1.280

THINNER COVERAGE BELOW

03 **Default model switched to Opus on more plans** BREAKING 55

Changes the default model on Pro and Team Standard plans from Sonnet to Opus, matching Max, Team Premium, and Enterprise plans; workloads relying on the previous Sonnet default will now use Opus 5.5 (`claude-opus-5-5`) and incur different token costs.

— Explains before/after but no migration path given v2.1.280

04 **Expanded hook execution telemetry** IMPROVED 50

Adds hook output sizes and the count of oversized outputs saved to a file to the `hook_execution_complete` OpenTelemetry event.

— Names the OTel event but no config or query example v2.1.280

05 **Mouse navigation in fullscreen lists** IMPROVED 48

Adds mouse support to more fullscreen lists: the scroll wheel navigates the `/skills` list, and skill state options in `/plugin` are now clickable.

— Names the affected UI commands, straightforward to try v2.1.280

└──▷ **BREAKING ON UPGRADE**

! The default model on Pro and Team Standard plans has changed from Sonnet to Opus; workloads relying on the previous Sonnet default will now use Opus 5.5 (`claude-opus-5-5`) and incur different token costs.

v2.1.280 NOTES ↗ CODE ↗ LATEST STABLE

Claude Code v2.1.280 adds Claude Opus 5.5 (1M context, `claude-opus-5-5`), `CLAUDE_CODE_MAX_MCP_DESCRIPTION_LENGTH`, and richer `hook_execution_complete` OTel telemetry.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.1.280 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.280
```

└──▷ **TRY IT**

Allow verbose MCP servers whose tool descriptions exceed the 2,048-character default to load fully, without truncation, in your Claude Code session.

```
$ CLAUDE_CODE_MAX_MCP_DESCRIPTION_LENGTH=8192 claude
```

Adds `CLAUDE_CODE_MAX_MCP_DESCRIPTION_LENGTH` environment variable to override the 2,048-character cap on MCP tool descriptions and server instructions for every MCP server in the session.

Adds Claude Opus 5.5 (`claude-opus-5-5`) as the new default Opus model, with 1M context window and $4/$20 per Mtok input/output pricing ($0.20/Mtok cache reads).

Adds hook output sizes and the count of oversized outputs saved to a file to the `hook_execution_complete` OpenTelemetry event.

Adds mouse support to more fullscreen lists: the scroll wheel navigates the `/skills` list, and skill state options in `/plugin` are now clickable.

Changes the default model on Pro and Team Standard plans from Sonnet to Opus, matching Max, Team Premium, and Enterprise plans.

└─▷ **BREAKING ON UPGRADE**

! The default model on Pro and Team Standard plans has changed from Sonnet to Opus; workloads relying on the previous Sonnet default will now use Opus 5.5 (`claude-opus-5-5`) and incur different token costs.

WAS THIS USEFUL?  

All Hands AI OpenHands

Feedback ⓘ v1.22.0 NOTES ↗

RANK

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

> OpenHands v1.22.0 lets users test remote MCP servers on cloud backends and extends Git Sync access to organization admins on cloud backends.

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

01 **Remote MCP server testing on cloud backends** NEW 35

Enables testing of remote MCP servers on cloud backends via the app server.

— Names the surface but no command or UI steps given. v1.22.0

02 **Git Sync access for org admins** IMPROVED 27

Git Sync is now opened to org admins on cloud backends.

— Only a one-line summary mention, no mechanism or path. v1.22.0

v1.22.0 NOTES ↗ LATEST STABLE

OpenHands v1.22.0 adds remote MCP server testing and opens Git Sync to org admins on cloud backends.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.22.0
https://github.com/OpenHands/OpenHands.git
# already have the repo? check out this version:
$ git checkout v1.22.0
```

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ desktop-v0.24.4 NOTES ↗

RANK

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

> Qwen Code's latest releases focus on remote access and execution security: expiring QR pairing and a `--token-qr` fallback for `qwen serve`, daemon-free SSH workspaces, tool-execution-scoped `bwrap` sandboxing, encrypted mobile connection profiles, and JDBC-backed managed tool execution in the Java SDK.

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

01 QR pairing and --token-qr flag for qwen serve NEW

88

Adds a `--token-qr` flag and an address-only QR fallback to `qwen serve` so clients can authenticate when the token QR cannot be displayed, and enables expiring QR pairing on non-loopback listeners in the web-shell so remote devices pair via a time-limited QR code instead of a persistent token URL.

Start the web server on a network-facing address with a time-limited QR code for pairing, rather than a persistent token URL.

```
$ qwen serve --token-qr
```

Serve Qwen Code over a non-loopback interface and display a token QR code when the normal pairing flow is unavailable — useful when running headless on a remote box.

```
$ qwen serve --open --token-qr
```

— Names exact flag and gives a runnable serve command. v0.24.4 desktop-v0.24.4

02 Managed tool execution infrastructure in Java SDK and CLI

NEW

60

Adds managed tool execution state and a runtime broker service core to the Java SDK, persists managed tool executions with JDBC in the Java SDK for durable execution state across restarts, and adds a managed runtime attestation contract to the CLI.

— Names JDBC persistence and attestation contract, no API signatures. desktop-v0.24.4

THINNER COVERAGE BELOW

03 bwrap sandboxing scoped to tool execution IMPROVED

55

Moves `bwrap` sandboxing from session level to the tool-execution layer, scoping Linux namespace isolation to individual tool calls rather than the whole session.

— Explains mechanism but no flag or config to invoke it. v0.24.4 desktop-v0.24.4

04 **Encrypted connection profiles for mobile** NEW 40

Adds encrypted, isolated connection profiles for mobile (Phase 2), storing per-connection credentials in a protected store.

— Notes phase and storage mechanism but no UI or API path. desktop-v0.24.4

05 **Plan drift tracking in /review skill** IMPROVED 40

Records what a plan was computed from and reports drift in the `/review` skill.

— Names the skill but no example of the drift output. desktop-v0.24.4

06 **Daemon-free SSH workspaces** NEW 35

Supports SSH workspaces without requiring a remote daemon, enabling agent connections to plain SSH hosts.

— States capability but gives no command or setup steps. desktop-v0.24.4

07 **Batched workspace session catalogs in daemon** IMPROVED 30

Adds batched workspace session catalogs to the daemon, improving multi-workspace session management.

— Vague improvement, no interface or config detail. v0.24.4

08 **Opt-in ACP child heap enforcement** NEW 25

Adds opt-in fixed ACP child heap enforcement on the serve daemon.

— Bare description, no mechanism or limit named. desktop-v0.24.4

09 **Monitor tool in system prompt guidance** NEW 25

Adds a monitor tool to system prompt guidance for improved agent observability.

— No specifics on what the monitor tool actually reports. v0.24.4

desktop-v0.24.4 NOTES ↗ LATEST STABLE

Qwen Code desktop-v0.24.4 adds SSH daemon-free workspaces, expiring QR pairing, `--token-qr` escape hatch, bwrap sandboxing at tool execution, ACP heap enforcement, encrypted mobile connection profiles, and JDBC-backed managed tool execution.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v0.24.4 https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout desktop-v0.24.4
```

└──▷ TRY IT

Serve Qwen Code over a non-loopback interface and display a token QR code when the normal pairing flow is unavailable — useful when running headless on a remote box.

```
$ qwen serve --open --token-qr
```

Adds `--token-qr` flag and address-only QR fallback to `qwen serve` for authenticating clients when the token QR cannot be displayed.

Moves `bwrap` sandboxing to tool execution level, scoping Linux namespace isolation to individual tool calls rather than the whole session.

Adds opt-in fixed ACP child heap enforcement on the serve daemon.

Supports SSH workspaces without requiring a remote daemon, enabling agent connections to plain SSH hosts.

Enables expiring QR pairing on non-loopback listeners in the web-shell, allowing remote devices to pair via a time-limited QR code.

+5 more

v0.24.4 NOTES ↗ STABLE

Qwen Code: Enables expiring QR pairing on non-loopback listeners in the web-shell, reducing the window of exposure for QR-based auth on network-facing servers.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.24.4 https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout v0.24.4
```

└──▷ TRY IT

Start the web server on a network-facing address with a time-limited QR code for pairing, rather than a persistent token URL.

```
$ qwen serve --token-qr
```

Adds batched workspace session catalogs to the daemon, improving multi-workspace session management.

Moves `bwrap` sandboxing to the tool-execution layer in the CLI.

Adds a monitor tool to system prompt guidance for improved agent observability.

WAS THIS USEFUL? 👍 👎

DeepSeek Harness

Feedback ⓘ dsh-v0.1.7-rc.1 CODE ↗ RANK

DeepSeek Harness is an open-source agent harness for building coding agents with plugins.

> DeepSeek Harness shipped mirror-based recovery for plugin and voice-input downloads, a profile-local compatibility permissions file, and a new tool-execution-stage UI, alongside configurable keyboard shortcuts, desktop DevTools access, live plugin Config schema inspection, and a batch of chat/panel UI refinements.

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

01 **Profile-local compatibility permissions file** NEW 76

Adds a `PROFILE_COMPATIBILITY_FILENAME` (`compatibility.json`) profile-local compatibility permissions file, readable before any Cordis plugins load, to control per-profile plugin version exemptions.

— Names the exact file and its load-order guarantee, but no usage walkthrough

dsh-v0.1.7-rc.1

02 **Desktop DevTools shortcut** NEW 70

Adds `F12` (Windows/Linux) and `Command Option I` (macOS) to open renderer DevTools in packaged desktop builds.

— Exact key combinations a reader can use immediately dsh-v0.1.7-alpha.2

THINNER COVERAGE BELOW

03 **Configurable keyboard shortcuts** NEW 55

Adds configurable Web and desktop keyboard shortcuts, including workspace/session commands, focused panel/page commands, and fixed stop and approval actions.

— Categorizes shortcut types but doesn't list actual keys or config location

dsh-v0.1.7-alpha.2

04 **Mirror fallback for plugin and voice-input downloads** NEW 54

The plugin manager now falls back to a mirror when GitHub is unreachable during plugin installation, and voice-input model preparation gains download mirror selection via `feat(voice-input): expose preparation download source selection`.

— Names the mirror mechanism and a commit-style flag but no manual trigger

dsh-v0.1.7-rc.1

05 **Live plugin Config schema inspection via tool-cordis** NEW

50

Adds live plugin Config schema inspection via `tool-cordis` (#4834).

— Names the tool and issue but no usage detail beyond that dsh-v0.1.7-alpha.2

06 **Tool execution stage UI in chat** NEW 47

The chat UI now presents tools across preparing, start, and result stages, giving practitioners live visibility into tool execution lifecycle.

— Describes the three stages but not how to invoke or configure it dsh-v0.1.7-rc.1

07 **Chat and panel UI refinements** IMPROVED 46

The agent-team Web Team panel gains adaptive header interactions; the preview panel unifies document zoom controls; the chat view adds grouped work details and stable history paging; and the web UI unifies code block styling and icon controls.

— Names four UI areas touched but only cosmetic detail, no config or steps

dsh-v0.1.7-alpha.2 dsh-v0.1.7-rc.1

08 **Automatic session stop before archiving** IMPROVED 38

Adds automatic stopping of a Session's running work before archiving it (#4765).

— States the behavior change but no mechanism or trigger details dsh-v0.1.7-alpha.2

09 **Progressive creator skills in agent preset** NEW 37

Adds progressive creator skills and profile shell facts to the agent preset (#4836).

— Names the preset addition but not what the skills or facts do dsh-v0.1.7-alpha.2

10 **Volatile Config projection in settings** NEW 37

Adds volatile Config projection through profile-backed forms in settings (#4587).

— Names the mechanism but not its effect or how to access it dsh-v0.1.7-alpha.2

11 **Context initialization from transient start matches** NEW

30

Conversation contexts can now be initialized from transient start matches, enabling new session-bootstrap paths.

— Bare description of a bootstrap path with no mechanism or entry point dsh-v0.1.7-rc.1

dsh-v0.1.7-rc.1 CODE ↗ PRE-RELEASE

DeepSeek Harness rc.1 adds mirror recovery for plugin installs, voice-input mirror selection, richer tool-stage UI, and context initialization from transient matches.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch dsh-v0.1.7-rc.1 https://github.com/deepseek-ai/deepseek-harness.git
# already have the repo? check out this version:
$ git checkout dsh-v0.1.7-rc.1
```

Adds mirror recovery fallback for GitHub plugin installation failures in the plugin manager, so installs can succeed when GitHub is unreachable.

Adds download mirror selection for voice-input model preparation via `feat(voice-input): expose preparation download source selection`.

Adds a `PROFILE_COMPATIBILITY_FILENAME` (`compatibility.json`) profile-local compatibility permissions file, readable before any Cordis plugins load, to control per-profile plugin version exemptions.

Presents tools in preparing, start, and result stages in the chat UI, giving practitioners live visibility into tool execution lifecycle.

Initializes conversation contexts from transient start matches, enabling new session-bootstrap paths.

+1 more

dsh-v0.1.7-alpha.2 CODE ↗ PRE-RELEASE

DeepSeek Harness v0.1.7-alpha.2 adds configurable keyboard shortcuts, session archiving, DevTools access, and live plugin config inspection.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch dsh-v0.1.7-alpha.2 https://github.com/deepseek-ai/deepseek-harness.git
# already have the repo? check out this version:
$ git checkout dsh-v0.1.7-alpha.2
```

Adds configurable Web and desktop keyboard shortcuts, including workspace/session commands, focused panel/page commands, and fixed stop and approval actions.

Adds `F12` (Windows/Linux) and `Command Option I` (macOS) to open renderer DevTools in packaged desktop builds.

Adds live plugin Config schema inspection via `tool-cordis` (#4834).

Adds automatic stopping of a Session's running work before archiving it (#4765).

Adds progressive creator skills and profile shell facts to the agent preset (#4836).

> +4 more

WAS THIS USEFUL? 👍 👎

Cotool

Feedback ⓘ v0.70.0 NOTES ↗ RANK

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

> Cotool v0.70.0 adds new Fleet and Azure CLI integrations, Grok 4.6 model support, per-repository GitHub App routing, and expands hunt-intel, Alerts, and Push Security investigation capabilities alongside agent administration improvements.

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

01 Azure CLI integration with workload identity federation NEW

75

Adds an Azure CLI integration with workload identity federation so agents can investigate Azure tenants without stored client secrets. Configured via Integrations > Azure CLI in Cotool.

Investigate Azure tenant resources from an agent without storing client secrets by using workload identity federation

> 📍 In Cotool, go to Integrations › Azure CLI and configure the workload identity federation settings for your Azure tenant — no client secret storage required.

— Names mechanism and exact UI path to configure it. v0.70.0

02 Per-repository GitHub App routing NEW

75

Adds per-repository routing across Cotool-managed and bring-your-own GitHub Apps, with conflict detection when both cover the same organization. Routing rules are configured in Settings > GitHub Apps, and Cotool flags any repository covered by more than one App.

Route specific repositories to different GitHub Apps to avoid conflicts when both a Cotool-managed and a bring-your-own App cover the same organization

> 📍 In Cotool, go to Settings › GitHub Apps and configure per-repository routing rules; Cotool will flag any repository covered by more than one App.

— Names the conflict behaviour and exact UI location. v0.70.0

03 Expanded /hunt-intel hunting sources IMPROVED

75

Expands `/hunt-intel` to hunt attached files, pasted reports, and hosted documents while ensuring explicit submissions always produce a hunt. Invoked in the chat interface with `/hunt-intel` plus pasted text or an attached file.

Submit a pasted threat report directly to hunt-intel to ensure it always triggers a full hunt

> 📍 In the Cotool chat interface, invoke `/hunt-intel` and paste the report text or attach a file — explicit submissions are guaranteed to produce a hunt.

— Names the exact command and its new input types. v0.70.0

THINNER COVERAGE BELOW

04 **Agent administration tag and settings improvements** IMPROVED 50

Improves agent administration with bulk tag management, customizable tag colors, and consolidated output, delivery, and failure-notification settings.

— Names three admin improvements but no exact controls or paths. v0.70.0

05 **Fleet integration for host investigation** NEW 45

Adds a Fleet integration for investigating hosts, policies, software, vulnerabilities, device activity, and live queries.

— Lists investigation areas but gives no setup steps or config. v0.70.0

06 **Expanded Push Security inspection** IMPROVED 45

Expands Push Security to inspect browser-control configurations, blocked URLs, and custom detections.

— Lists new inspection targets without usage steps. v0.70.0

07 **Ungrouped table view in Alerts** NEW 40

Adds an ungrouped table view to Alerts with sortable detection names and more room for alert titles.

— Describes the UI addition but no navigation path given. v0.70.0

08 **Detection proposal review controls** NEW 35

Adds detection proposal review controls to Hunt alerts and Slack notifications.

— Names surfaces affected but not how review works. v0.70.0

09 **Grok 4.6 added to model catalog** NEW 20

Adds Grok 4.6 to the model catalog for coding and agent workflows.

— Bare mention of a new model with no usage detail. v0.70.0

v0.70.0 NOTES ↗ LATEST STABLE

Cotool v0.70.0 adds Fleet, Azure CLI, and Grok 4.6 integrations plus per-repository GitHub App routing

└──▷ **HOW TO FIND IT**

Investigate Azure tenant resources from an agent without storing client secrets by using workload identity federation

> 📍 In Cotool, go to Integrations › Azure CLI and configure the workload identity federation settings for your Azure tenant — no client secret storage required.

Submit a pasted threat report directly to hunt-intel to ensure it always triggers a full hunt

> 📍 In the Cotool chat interface, invoke `/hunt-intel` and paste the report text or attach a file — explicit submissions are guaranteed to produce a hunt.

Route specific repositories to different GitHub Apps to avoid conflicts when both a Cotool-managed and a bring-your-own App cover the same organization

> 📍 In Cotool, go to Settings › GitHub Apps and configure per-repository routing rules; Cotool will flag any repository covered by more than one App.

Adds a Fleet integration for investigating hosts, policies, software, vulnerabilities, device activity, and live queries

Adds an Azure CLI integration with workload identity federation so agents can investigate Azure tenants without stored client secrets

Adds Grok 4.6 to the model catalog for coding and agent workflows

Adds per-repository routing across Cotool-managed and bring-your-own GitHub Apps, with conflict detection when both cover the same organization

Expands `/hunt-intel` to hunt attached files, pasted reports, and hosted documents while ensuring explicit submissions always produce a hunt

+4 more

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.32.18 NOTES ↗ RANK

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

> Letta Code v0.32.18 introduces a new MemFS v2 root layout for cloud agents alongside several under-the-hood memory and TUI reliability improvements.

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

01 **MemFS v2 root layout for cloud agents** NEW 50

New cloud agents are now created on the `MemFS v2` root layout, updating the default filesystem structure used for cloud-backed agents.

– Names the new layout but no migration or config detail given v0.32.18

02 **TUI thinking block collapse** IMPROVED 45

The terminal UI now hides system reminders and collapses thinking blocks by default, reducing noise during interactive sessions.

– Clear behavior change but no toggle or flag named v0.32.18

03 **Serialized writers for memory checkout** IMPROVED 40

Harness writers of a memory checkout are now serialized, enabling consistent state across concurrent memory operations.

– Names the mechanism but no measurable scope or config v0.32.18

04 **Preserved input IDs in subagent delegation** IMPROVED 35

Initial input IDs are now preserved in subagent delegation, maintaining message traceability across agent hierarchies.

– States the fix but no example of resulting traceability v0.32.18

05 **Background memory upkeep worker** IMPROVED 30

Delegated memory upkeep now runs in a silent background worker, keeping memory maintenance off the critical path of interactive sessions.

– Describes benefit but no mechanism or config specifics v0.32.18

v0.32.18 NOTES ↗ LATEST STABLE

> Letta Code v0.32.18 adds MemFS v2 root layout for cloud agents, TUI thinking collapse, and background memory upkeep.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.32.18 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.32.18
```

Creates new cloud agents on the `MemFS v2` root layout, updating the default filesystem structure for cloud-backed agents.

Hides system reminders and collapses thinking blocks by default in the TUI, reducing noise during interactive sessions.

Runs delegated memory upkeep in a silent background worker, keeping memory maintenance off the critical path.

Serializes harness writers of a memory checkout, enabling consistent state across concurrent memory operations.

Preserves initial input IDs in subagent delegation, maintaining message traceability across agent hierarchies.

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.7.1 NOTES ↗

RANK

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

> Zeroshot v10.7.1 introduces a built-in `auto-research` graph that runs bounded scout, experiment, judge, and decision iterations with persistent state and optional checkpointing.

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

01 **Built-in auto-research graph with bounded iteration cycles**

NEW 88

The new `auto-research` graph runs bounded scout, experiment, judge, and decision iterations to investigate a goal without manual looping. It persists a research charter, ledger, retained-workspace status, and historical findings with reversible experiment handling across runs, and supports optional per-iteration push checkpoints. Launch it with `zeroshot run --title <title> --template auto-research --input ./input.json --runtime-config ./runtime.json`, and inspect its iteration graph and checkpoint options via `zeroshot template show auto-research`.

Run the new auto-research graph against a goal to execute bounded scout-experiment-judge-decision cycles without manual iteration.

```
$ zeroshot run \
  --title 'Research async Rust runtimes for embedded targets' \
  --template auto-research \
  --input ./input.json \
  --runtime-config ./runtime.json
```

Inspect the auto-research template structure before launching a run to understand its iteration graph and checkpoint options.

```
$ zeroshot template show auto-research
```

— Names commands, template, and persisted state but not internal judge logic. v10.7.1

v10.7.1 NOTES ↗ LATEST STABLE

> Zeroshot v10.7.1 adds a built-in `auto-research` graph with bounded scout, experiment, judge, and decision iterations.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v10.7.1 https://github.com/the-open-engine/zeroshot.git
# already have the repo? check out this version:
$ git checkout v10.7.1
```

└──▷ **TRY IT**

Run the new auto-research graph against a goal to execute bounded scout-experiment-judge-decision cycles without manual iteration.

```
$ zeroshot run \
  --title 'Research async Rust runtimes for embedded targets' \
  --template auto-research \
  --input ./input.json \
  --runtime-config ./runtime.json
```

Inspect the auto-research template structure before launching a run to understand its iteration graph and checkpoint options.

```
$ zeroshot template show auto-research
```

Persists a research charter, ledger, retained-workspace status, and historical findings with reversible experiment handling across `auto-research` runs.

Supports optional per-iteration push checkpoints in the `auto-research` graph.

WAS THIS USEFUL? 👍 👎

Trailhq Graft

Feedback ⓘ v0.19.0 CODE ↗ RANK

Graft builds a code knowledge graph that gives coding agents compact, repository-specific context through a CLI and MCP server.

> Graft v0.19.0 lets coding agents and CI read public GitHub repositories without a GitHub App installation, and makes `graft brain push` hold the terminal to report live build progress.

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

01 Public repo brain reads without App installation NEW

91

Graft now supports reading public GitHub repositories via `POST /brain/build` without requiring the GitHub App to be installed: a failed installation lookup now attempts an open read instead of raising `RepoNotAccessibleError`. A new `GRAFT_PUBLIC_GITHUB_TOKEN` environment variable supplies a token so these reads borrow a real installation token rather than falling back to anonymous access, which is capped at 60 requests per hour; new `publicToken` and `publicOwner` fields on `BrainBuildDeps` carry this token through the brain builder.

Supply a GitHub token so public-repo brain builds borrow a real installation token instead of hitting the 60-req/hr anonymous ceiling — useful in CI where anonymous exhaustion would silently stall builds.

```
$ GRAFT_PUBLIC_GITHUB_TOKEN=ghp_yourtoken graft brain push
```

— Names env var, fields, endpoint, error, and rate limit. v0.19.0

THINNER COVERAGE BELOW

02 Terminal-held brain push with live stage reporting IMPROVED

52

`graft brain push` now waits for the brain build to complete instead of returning immediately after queuing, reporting the same four build stages shown in the browser and handing the prompt back after the first (mining) slice.

— Describes behavior change but stage names left unspecified. v0.19.0

> v0.19.0 CODE ↗ LATEST STABLE
>
> Graft v0.19.0 adds public-repo brain reads without App installation and terminal-held brain pushes with live stage reporting.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.19.0 https://github.com/trailhq/Graft.git
# already have the repo? check out this version:
$ git checkout v0.19.0
```

└──▷ **TRY IT**

Supply a GitHub token so public-repo brain builds borrow a real installation token instead of hitting the 60-req/hr anonymous ceiling — useful in CI where anonymous exhaustion would silently stall builds.

```
$ GRAFT_PUBLIC_GITHUB_TOKEN=ghp_yourtoken graft brain push
```

Adds `GRAFT_PUBLIC_GITHUB_TOKEN` environment variable to supply a token for reading public repositories the GitHub App is not installed on, avoiding the 60-requests-per-hour anonymous rate limit.

Adds `publicToken` and `publicOwner` fields to `BrainBuildDeps` so the brain builder can borrow an installation token for public-repo reads instead of falling back to anonymous access.

Supports reading public GitHub repositories via `POST /brain/build` without requiring the GitHub App to be installed — a failed installation lookup now attempts an open read rather than raising `RepoNotAccessibleError`.

Adds `graft brain push` terminal-hold behaviour: the push now waits for the brain build to complete, reports the same four build stages shown in the browser, and hands the prompt back after the first (mining) slice rather than immediately after queuing.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

PydanticAI

Feedback ⓘ v2.48.0 NOTES ↗ RANK

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

> PydanticAI v2.48.0 expands model support with new OpenAI and Anthropic models, including audio-capable OpenAI models.

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

01 **New OpenAI and Anthropic model support** NEW 65

Adds support for OpenAI `gpt-6-sol` and `gpt-6-luna`, Anthropic `claude-opus-5-5`, and new OpenAI audio models via `genai-prices` 0.1.7.

Target the new GPT-6 Sol model for a one-shot prompt from the terminal.

```
$ clai --model openai:gpt-6-sol 'Summarize the OWASP Top 10 for 2025'
```

Use Claude Opus 5.5 as the backbone of a PydanticAI agent for high-complexity reasoning tasks.

```python
from pydantic_ai import Agent

agent = Agent('anthropic:claude-opus-5-5')
result = agent.run_sync('Analyze this incident report and identify root causes.')
print(result.output)
```

— Names exact model identifiers and shows runnable commands, but no mechanism detail.

v2.48.0

v2.48.0 NOTES ↗ LATEST STABLE

PydanticAI v2.48.0 adds OpenAI `gpt-6-sol`, `gpt-6-luna`, and Anthropic `claude-opus-5-5` model support plus new OpenAI audio models.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.48.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.48.0
```

└──▷ **TRY IT**

Target the new GPT-6 Sol model for a one-shot prompt from the terminal.

```
$ clai --model openai:gpt-6-sol 'Summarize the OWASP Top 10 for 2025'
```

Use Claude Opus 5.5 as the backbone of a PydanticAI agent for high-complexity reasoning tasks.

```python
from pydantic_ai import Agent

agent = Agent('anthropic:claude-opus-5-5')
result = agent.run_sync('Analyze this incident report and identify root causes.')
print(result.output)
```

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ langchain-anthropic==1.7.3 NOTES ↗ RANK

LangChain is an open-source framework that orchestrates applications powered by language models.

> langchain-anthropic 1.7.3 improves multi-turn message handling and simplifies structured output for newer Claude models.

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

01 Auto-routed structured output for new Claude models IMPROVED

75

`with_structured_output` now auto-routes to `method="json_schema"` for Claude fable and opus 5.5 models, removing the need to manually specify the method for these models.

Extract structured output from Claude opus 5.5 without manually specifying `method='json_schema'` — the library now selects it automatically.

python

```python
from langchain_anthropic import ChatAnthropic
from pydantic import BaseModel

class Summary(BaseModel):
    title: str
    key_points: list[str]

llm = ChatAnthropic(model="claude-opus-5-5")
structured_llm = llm.with_structured_output(Summary)
result = structured_llm.invoke("Summarize the benefits of zero-trust networking.")
print(result)
```

— Names exact method and models plus runnable example langchain-anthropic==1.7.3

THINNER COVERAGE BELOW

02 In-place mid-conversation SystemMessage handling IMPROVED

50

Anthropic message construction now sends mid-conversation `SystemMessage`s in place rather than hoisting them to the top of the message list, enabling accurate multi-turn conversation construction with Anthropic models.

— Explains behavior change but no code example given langchain-anthropic==1.7.3

`langchain-anthropic==1.7.3` NOTES ↗ LATEST STABLE

langchain-anthropic 1.7.3 adds in-place mid-conversation SystemMessage support and auto-routes structured output for new Anthropic models.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch langchain-anthropic==1.7.3
https://github.com/langchain-ai/langchain.git
# already have the repo? check out this version:
$ git checkout langchain-anthropic==1.7.3
```

└──▷ **USE IT**

Extract structured output from Claude opus 5.5 without manually specifying `method='json_schema'` — the library now selects it automatically.

python

```python
from langchain_anthropic import ChatAnthropic
from pydantic import BaseModel

class Summary(BaseModel):
    title: str
    key_points: list[str]

llm = ChatAnthropic(model="claude-opus-5-5")
structured_llm = llm.with_structured_output(Summary)
result = structured_llm.invoke("Summarize the benefits of zero-trust
networking.")
print(result)
```

Sends mid-conversation `SystemMessage`s in place rather than hoisting them to the top of the message list, enabling accurate multi-turn conversation construction with Anthropic models.

Auto-routes `with_structured_output` to `method="json_schema"` for Claude fable and opus 5.5 models, removing the need to manually specify the method for these models.

WAS THIS USEFUL? 👍 👎

Agno (formerly Phidata)

Feedback ⓘ v3.0.11 NOTES ↗

RANK

Agno is an agent framework and runtime that orchestrates and runs multi-agent systems.

> Agno v3.0.11 reworks knowledge retrieval around a new knowledge-level pipeline with pluggable rerankers (MMR and Recency), adds documentation-source migration tools, typed page tool results, a machine-readable cancellation state, and a new YAPI model provider.

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

01 Machine-readable cancellation stage on run outputs NEW

84

Adds a machine-readable `cancellation_stage` field (`PENDING`, `EXECUTING`, `INTERRUPTED`, or unknown) to `RunOutput`, `TeamRunOutput`, `WorkflowRunOutput`, and the run API schemas, letting automation branch on cancellation state instead of matching on message text.

Detect cancellation state programmatically instead of matching on message text, so automation can branch on `PENDING`, `EXECUTING`, or `INTERRUPTED`.

python

```
run_output = agent.run('scan all endpoints')
if run_output.cancellation_stage == 'INTERRUPTED':
    print('Run was interrupted mid-execution — review partial
findings')
elif run_output.cancellation_stage == 'PENDING':
    print('Run cancelled before it started')
```

— Named field, enum values, and a runnable example provided. v3.0.11

02 Recency reranker weights newer document revisions NEW

80

Adds `RecencyReranker`, which blends the search score with an exponential decay on a timestamp so newer revisions outrank superseded ones, focused on pgvector.

Use RecencyReranker so that the most recently indexed document revisions outrank older superseded ones in pgvector-backed knowledge bases.

```python
from agno.knowledge import Knowledge
from agno.reranker import RecencyReranker

knowledge = Knowledge(
    vector_db=my_pgvector_db,
    reranker=RecencyReranker(),
)
results = knowledge.search('CVE patch advisory')
```

— Mechanism and vector-db scope given plus runnable example. v3.0.11

03 **Knowledge-level retrieval pipeline replaces per-vector-db reranking**

75

BREAKING

Knowledge.search() now runs a knowledge-level retrieval pipeline that widens the candidate pool before a reranker reorders it, so rerankers no longer need to be implemented per vector db. The `reranker` parameter on vector databases is deprecated in favour of the reranker on `Knowledge`; existing per-vector-db reranker configurations will need to be moved to the `Knowledge` `-level` reranker.

— Explains mechanism and migration path but gives no exact migration steps. v3.0.11

04 **MMR reranker for diverse retrieval results** NEW

75

Adds `MMRReranker` (Maximal Marginal Relevance) as a retrieval strategy that balances relevance against diversity so near-duplicate chunks do not crowd out the result set.

Attach an MMR reranker to a Knowledge base to surface diverse results and suppress near-duplicate chunks at query time.

```python
from agno.knowledge import Knowledge
from agno.reranker import MMRReranker

knowledge = Knowledge(
    vector_db=my_vector_db,
    reranker=MMRReranker(),
)
results = knowledge.search('zero-day exploit techniques',
num_documents=10)
```

— Named class with mechanism and runnable usage example. v3.0.11

05 **Documentation source migration tools in Knowledge** NEW

73

Adds `Knowledge.inspect_page_source` / `ainspect_page_source` and guarded `migrate_page_source` / `amigrate_page_source` to move an indexed documentation source to a new hostname; migration is a dry run by default.

— Names all four methods and dry-run default, but no example. v3.0.11

06 **Typed page command results and ranked search tool output** NEW

68

Adds `PageCommandResult` and `PageFileSystem.run_command_result` / `arun_command_result` with explicit error, completeness, and truncation metadata; Knowledge.get_tools(page_results=True) returns ranked `SearchResult` objects, including as MCP structured output.

— Names classes and flag but lacks a usage example. v3.0.11

THINNER COVERAGE BELOW

07 **Optional previous_response_id chaining for OpenAIResponses**

IMPROVED

55

Adds `use_previous_response_id` to `OpenAIResponses` so `store=True` no longer forces automatic `previous_response_id` chaining.

— Names the exact flag but gives no usage example. v3.0.11

08 **DeepSeek thinking-mode models recognized via OpenAILike**

IMPROVED

45

DeepSeek thinking-mode model IDs are now recognised as reasoning models when served through any `OpenAILike` provider such as `DashScope`.

— Names providers involved but no config or example given. v3.0.11

09 **YAPI model provider support** NEW

33

Adds `YAPI` as an OpenAI-compatible model provider.

— Bare name with no configuration or usage detail. v3.0.11

└──▷ **BREAKING ON UPGRADE**

! The `reranker` parameter on vector databases is deprecated in favour of the reranker on `Knowledge`; existing per-vector-db reranker configurations will need to be moved to the `Knowledge` `-level` reranker.

> v3.0.11 NOTES ↗ LATEST STABLE
>
> Agno v3.0.11 adds MMR and Recency rerankers, a knowledge-level retrieval pipeline, `cancellation_stage`, typed page tool results, and the YAPI model

provider.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v3.0.11 https://github.com/agno-agi/agno.git
# already have the repo? check out this version:
$ git checkout v3.0.11
```

└──▷ **USE IT**

Attach an MMR reranker to a Knowledge base to surface diverse results and suppress near-duplicate chunks at query time.

```python
from agno.knowledge import Knowledge
from agno.reranker import MMRReranker

knowledge = Knowledge(
    vector_db=my_vector_db,
    reranker=MMRReranker(),
)
results = knowledge.search('zero-day exploit techniques',
num_documents=10)
```

Use RecencyReranker so that the most recently indexed document revisions outrank older superseded ones in pgvector-backed knowledge bases.

```python
from agno.knowledge import Knowledge
from agno.reranker import RecencyReranker

knowledge = Knowledge(
    vector_db=my_pgvector_db,
    reranker=RecencyReranker(),
)
results = knowledge.search('CVE patch advisory')
```

Detect cancellation state programmatically instead of matching on message text, so automation can branch on `PENDING`, `EXECUTING`, or `INTERRUPTED`.

```python
run_output = agent.run('scan all endpoints')
if run_output.cancellation_stage == 'INTERRUPTED':
    print('Run was interrupted mid-execution — review partial
findings')
elif run_output.cancellation_stage == 'PENDING':
    print('Run cancelled before it started')
```

Adds `MMRReranker` (Maximal Marginal Relevance) as a retrieval strategy that balances relevance against diversity so near-duplicate chunks do not crowd out the result set.

Adds `RecencyReranker`, which blends the search score with an exponential decay on a timestamp so newer revisions outrank superseded ones, focused on pgvector.

Adds `Knowledge.inspect_page_source` / `ainspect_page_source` and guarded `migrate_page_source` / `amigrate_page_source` to move an indexed documentation source to a new hostname; migration is a dry run by default.

Adds `PageCommandResult` and `PageFileSystem.run_command_result` / `arun_command_result` with explicit error, completeness, and truncation metadata; Knowledge.get_tools(page_results=True) returns ranked `SearchResult` objects, including as MCP structured output.

Adds machine-readable `cancellation_stage` field (`PENDING`, `EXECUTING`, `INTERRUPTED`, or unknown) to `RunOutput`, `TeamRunOutput`, `WorkflowRunOutput`, and the run API schemas.

+4 more

└──▷ BREAKING ON UPGRADE

! The `reranker` parameter on vector databases is deprecated in favour of the reranker on `Knowledge`; existing per-vector-db reranker configurations will need to be moved to the `Knowledge` `-level` reranker.

WAS THIS USEFUL? 👍 👎

deepset Haystack

Feedback ⓘ v3.2.0-rc1 NOTES ↗ RANK

Haystack builds language-model applications with modular pipelines for retrieval-augmented generation and agents.

> Haystack v3.2.0-rc1 adds a fluent pipeline-building API, several new Agent hook controls (stop_run, TokenBudgetHook, SummarizationCompactor), token-based document splitting, and tightens Jinja sandbox and Toolset serialization rules with accompanying breaking changes.

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

01 **Token-based document splitting** NEW 95

Adds `split_by='token'` mode to `DocumentSplitter` and `DocumentPreprocessor`, splitting text by LLM token count via tiktoken; encoding defaults to `'o200k_base'` and is configurable via the new `tokenizer_encoding` parameter; `split_length`, `split_overlap`, and `split_threshold` all apply; requires `pip install tiktoken`.

Split documents by token count (instead of words/sentences) to match LLM context windows precisely.

python

```python
from haystack.components.preprocessors import DocumentSplitter

splitter = DocumentSplitter(
    split_by="token",
    split_length=512,
    split_overlap=32,
    tokenizer_encoding="o200k_base",
)
```

— Runnable example plus every parameter and default named. v3.2.0-rc1

02 **Fluent pipeline-building API** NEW 90

Adds Pipeline.add_components() to register a dict of component names to instances in a single call, with re-adding the same component under the same name now a no-op; adds Pipeline.connect_many() to wire multiple (sender, receiver) pairs at once; Pipeline.add_component() and Pipeline.add_components() now return the pipeline itself so calls can be chained.

Build a pipeline with a fluent chain of calls instead of separate add/connect statements.

```python
from haystack import Pipeline

pipeline = (
    Pipeline()
    .add_components(
        {
            "retriever": retriever,
            "prompt_builder": prompt_builder,
            "llm": llm,
        }
    )
    .connect_many(
        [
            ("retriever", "prompt_builder.documents"),
            ("prompt_builder", "llm"),
        ]
    )
)
```

— Named methods with a runnable chained example. v3.2.0-rc1

03 **Process-wide unsafe deserialization flag and Jinja sandbox requirement**

90

BREAKING

Adds the `HAYSTACK_UNSAFE_DESERIALIZATION` environment variable as a process-wide equivalent of `unsafe=True` for every `Pipeline.load` / `Pipeline.loads` / `Pipeline.from_dict`, `Tool.from_dict`, `State.from_dict`, and `ConditionalRouter` / `OutputAdapter` Jinja sandbox call, accepting truthy values `1` or `true` and read once then frozen for the process lifetime. As a consequence, serialized `OutputAdapter` and `ConditionalRouter` components containing Jinja `custom_filters` must now be loaded with Pipeline.load(..., unsafe=True) (or the equivalent `Pipeline.loads` / `Pipeline.from_dict` option).

— Names env var, values, and migration requirement clearly. v3.2.0-rc1

04 **TokenBudgetHook caps per-run token spend** NEW

90

Adds experimental `TokenBudgetHook` in `haystack.hooks.budget` to cap total token spend per Agent run, ending the run with `exit_reason='token_budget_exceeded'` when the limit is hit.

Cap an Agent run's total token spend so runaway loops cannot exhaust your budget.

```python
from haystack.components.agents import Agent
from haystack.components.generators.chat import
OpenAIChatGenerator
from haystack.hooks.budget import TokenBudgetHook

agent = Agent(
    chat_generator=OpenAIChatGenerator(model="gpt-4o-mini"),
    tools=[search],
    hooks={"before_llm":
[TokenBudgetHook(max_total_tokens=100_000)]},
)
```

— Runnable example with exact class and exit reason. v3.2.0-rc1

05 **Binary content offloading for tool results** BREAKING

85

Extends `ToolResultOffloadHook` to offload `ImageContent` and `FileContent` tool results (not just text), with each part getting its own store entry and binary content decoded from base64; opt-in via the new `supports_binary_content` attribute on the `ToolResultStore` protocol, which `FileSystemToolResultStore` sets to `True`. As a result, the `tool_result_offloaded` meta key now always holds a list of store references instead of a single reference string, so code reading that key must index into the list.

— Names the protocol attribute and migration impact on the meta key. v3.2.0-rc1

06 **Toolset combination API changes** BREAKING

85

Removes the `+` operator for combining Toolsets — pass them as a list instead, e.g. Agent(..., tools=[toolset_a, toolset_b]); pipelines serialized with a `+ -combined` Toolset (referencing the removed `_ToolsetWrapper` class) cannot be loaded and must be recreated with the list form. Toolset.add() now only accepts a single `Tool`, not another `Toolset`.

— Exact migration path and removed class named. v3.2.0-rc1

07 **SummarizationCompactor compresses Agent history** NEW

80

Adds experimental `SummarizationCompactor` (used with `CompactionHook`) that progressively summarizes long Agent conversation histories through four tiers — `historical_turns`, `historical_summaries`, `current_task_steps`, `current_task_summaries` — until a target token budget is reached, with a configurable `min_keep_steps`.

— Mechanism and tiers named in detail but no example given. v3.2.0-rc1

08 **stop_run control flag for Agent hooks** NEW 75

Adds a `stop_run` control flag for Agent hooks: a hook calls state.set('stop_run', '<reason>') to end an Agent run cleanly without raising an error, and the reason is reported in `exit_reason`.

— Exact call named but no full example shown. v3.2.0-rc1

09 **Skill store directory from environment secret** IMPROVED 65

Allows `FileSystemSkillStore`'s `skills_dir` to accept a `Secret` (e.g. Secret.from_env_var('SKILLS_DIR')) so the path can be sourced from an environment variable instead of being hard-coded.

— Names exact API and env-var pattern but no full example. v3.2.0-rc1

THINNER COVERAGE BELOW

10 **Minimum content length filter in DocumentCleaner** NEW 55

Adds the `min_content_length` parameter to `DocumentCleaner` to drop text documents shorter than the threshold (after whitespace trimming) from the output.

— Named parameter but no example or default given. v3.2.0-rc1

└──▷ **BREAKING ON UPGRADE**

! Serialized `OutputAdapter` and `ConditionalRouter` components that contain Jinja `custom_filters` must now be loaded with Pipeline.load(..., unsafe=True) (or the equivalent `Pipeline.loads` / `Pipeline.from_dict` option).

! The `tool_result_offloaded` meta key now always holds a list of store references instead of a single reference string; code reading that key must index into the list.

! The `+` operator for combining Toolsets is removed; pass Toolsets as a list instead — Agent(..., tools=[toolset_a, toolset_b]); pipelines serialized with a `+ -combined` Toolset (referencing the removed `_ToolsetWrapper` class) cannot be loaded and must be recreated with the list form.

! Toolset.add() now only accepts a single `Tool`, not another `Toolset`; to combine Toolsets, pass them as a list to `Agent`.

v3.2.0-rc1 NOTES ↗ PRE-RELEASE

Haystack v3.2.0-rc1 adds pipeline-building fluent API, token-budget and summarization hooks, and token-based document splitting

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v3.2.0-rc1 https://github.com/deepset-
ai/haystack.git
# already have the repo? check out this version:
$ git checkout v3.2.0-rc1
```

└──▷ **USE IT**

Build a pipeline with a fluent chain of calls instead of separate add/connect statements.

```python
from haystack import Pipeline

pipeline = (
    Pipeline()
    .add_components(
        {
            "retriever": retriever,
            "prompt_builder": prompt_builder,
            "llm": llm,
        }
    )
    .connect_many(
        [
            ("retriever", "prompt_builder.documents"),
            ("prompt_builder", "llm"),
        ]
    )
)
```

Cap an Agent run's total token spend so runaway loops cannot exhaust your budget.

```python
from haystack.components.agents import Agent
from haystack.components.generators.chat import OpenAIChatGenerator
from haystack.hooks.budget import TokenBudgetHook

agent = Agent(
    chat_generator=OpenAIChatGenerator(model="gpt-4o-mini"),
    tools=[search],
    hooks={"before_llm":
[TokenBudgetHook(max_total_tokens=100_000)]},
)
```

Split documents by token count (instead of words/sentences) to match LLM context windows precisely.

```python
from haystack.components.preprocessors import DocumentSplitter

splitter = DocumentSplitter(
    split_by="token",
    split_length=512,
    split_overlap=32,
    tokenizer_encoding="o200k_base",
)
```

Adds Pipeline.add_components() method to register a dict of component names to instances in a single call; re-adding the same component under the same name is now a no-op.

Adds Pipeline.connect_many() to wire multiple (sender, receiver) pairs at once; Pipeline.add_component() and Pipeline.add_components() now return the pipeline for method chaining.

Adds the `HAYSTACK_UNSAFE_DESERIALIZATION` environment variable as a process-wide equivalent of `unsafe=True` on every `Pipeline.load` / `Pipeline.loads` / `Pipeline.from_dict`, `Tool.from_dict`, `State.from_dict`, and `ConditionalRouter` / `OutputAdapter` Jinja sandbox call; accepts truthy values `1` or `true`; read once and frozen for the process lifetime.

Adds `split_by='token'` mode to `DocumentSplitter` and `DocumentPreprocessor`, splitting text by LLM token count via tiktoken; encoding defaults to `'o200k_base'` and is configurable via the new `tokenizer_encoding` parameter; `split_length`, `split_overlap`, and `split_threshold` all apply; requires `pip install tiktoken`.

Adds the `min_content_length` parameter to `DocumentCleaner` to drop text documents shorter than the threshold (after whitespace trimming) from the output.

+5 more

└──▷ **BREAKING ON UPGRADE**

! Serialized `OutputAdapter` and `ConditionalRouter` components that contain Jinja `custom_filters` must now be loaded with Pipeline.load(..., unsafe=True) (or the equivalent `Pipeline.loads` / `Pipeline.from_dict` option).

! The `tool_result_offloaded` meta key now always holds a list of store references instead of a single reference string; code reading that key must index into the list.

! The `+` operator for combining Toolsets is removed; pass Toolsets as a list instead — Agent(..., tools=[toolset_a, toolset_b]); pipelines serialized with a `+ -`

`combined` Toolset (referencing the removed `_ToolsetWrapper` class) cannot be loaded and must be recreated with the list form.

! Toolset.add() now only accepts a single `Tool`, not another `Toolset`; to combine Toolsets, pass them as a list to `Agent`.

WAS THIS USEFUL? 👍 👎

Mastra

Feedback ⓘ @mastra/core@1.68.0 NOTES ↗ RANK

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

> Mastra shipped a Langfuse-to-Mastra trace migration CLI, platform-injected system environment variables, organization-level deploy/outage alerts, programmatic skill validation, and a large core release adding an MCP v2 server rewrite, new Azure AI Search and Weaviate vector stores, token-budgeted memory trimming, and richer trace pagination with delta polling.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 **Langfuse trace history import command** NEW 95

`mastra traces langfuse` migrates existing trace history from Langfuse into a Mastra Platform project. It supports `--dry-run` to preview an import and write counts/warnings/skip reasons to `report.json` without uploading; `--from` / `--to` (ISO 8601) to select an import window up to 30 days (default: last 30 days); `--resume <import-id>` to continue an interrupted import; and `--yes` to skip the confirmation prompt for CI. Authentication uses `MASTRA_API_TOKEN` / `MASTRA_ORG_ID`, source config uses `LANGFUSE_PUBLIC_KEY`, `LANGFUSE_SECRET_KEY`, and `LANGFUSE_BASE_URL` (supporting US, Japan, HIPAA Cloud, or self-hosted regions), and import state (`manifest.json`, `traces.jsonl`, `report.json`) is stored under `~/.mastra/imports/traces/<target-project-id>/<import-id>/`. Imports are capped at 5 GiB prepared data and 4 MiB per-trace payload (oversized traces recorded with `trace_too_large`), and after upload a deterministic sample of up to 10 traces is read back to verify span IDs, parent links, names, span types, event flags, timestamps, and error presence.

Preview a Langfuse trace migration before committing — validate what will be imported and catch skip reasons without touching the destination project.

```
$ npx mastra traces langfuse --project my-project --from "2025-01-01" --to "2025-01-31" --dry-run
```

Run a headless import in CI using environment-variable auth, skipping the confirmation prompt — suitable for automated migration pipelines.

```
$ MASTRA_API_TOKEN=mastra-api-token MASTRA_ORG_ID=organization-id LANGFUSE_PUBLIC_KEY=langfuse-public-key LANGFUSE_SECRET_KEY=langfuse-secret-key npx mastra traces langfuse --project my-project --yes
```

Resume a previously interrupted or paused import to retry verification or upload without re-reading the source or re-uploading acknowledged traces.

```
$ npx mastra traces langfuse --resume 00000000-0000-0000-0000-000000000000 --project my-project
```

— Full flag set, env vars, limits and runnable commands given product docs

02 **Platform-injected system environment variables** BREAKING

85

Every deploy automatically receives `MASTRA_PROJECT_ID`, `MASTRA_ENVIRONMENT_ID`, `MASTRA_ENVIRONMENT_NAME`, `MASTRA_ENVIRONMENT_SLUG`, `MASTRA_PLATFORM_REGION`, `MASTRA_PLATFORM_ACCESS_TOKEN`, `MASTRA_PLATFORM_BUCKET_NAME`, and `MASTRA_WORKERS`, plus managed database variables resolved at deploy time and never stored in the project: `TURSO_DATABASE_URL` / `TURSO_AUTH_TOKEN` for Turso, `DATABASE_URL` for Neon, and `POSTGRES_URL` for Postgres. `mastra env vars pull <environment> --output <file>` pulls the merged variable set for an environment into a local file, with managed values written as name-only comments. Variables stored under any of the reserved names above are silently overridden at runtime by the platform-injected value, so a stored value never reaches the running service.

Inspect the full resolved variable set for your staging environment before deploying, to verify which values the runtime will actually see.

```
$ mastra env vars pull staging --output .env.staging
```

Read the injected platform region at runtime to conditionally configure region-aware clients inside a deploy.

```javascript
const region = process.env.MASTRA_PLATFORM_REGION; // 'US' or 'EU'
const endpoint = region === 'EU' ? 'https://eu.example.com' : 'https://us.example.com';
```

— Names every reserved variable and the pull command, with override caveat product docs

03 **Streamed thread-title generation events** NEW

75

`generateTitle.emitEvent: true` on Memory options makes run streams emit a `data-thread-title` chunk (with `threadId` and `title`) before the `finish`

chunk, letting HTTP/stream clients receive generated thread titles without polling; also accepts `minMessages` and `model` sub-options.

Stream a generated thread title to an HTTP client immediately after a run completes, without polling.

```typescript
const memory = new Memory({
  options: {
    generateTitle: {
      emitEvent: true,
      minMessages: 2,
    },
  },
});

for await (const chunk of stream.fullStream) {
  if (chunk.type === 'data-thread-title') {
    console.log(chunk.data.threadId, chunk.data.title);
  }
}
```

— Named config, chunk shape and sub-options with a runnable stream example

`@mastra/core@1.68.0`

04 **Paginated trace queries with delta polling** NEW 75

queryTraces() gains page-based pagination (`pagination: { page, perPage }`) and a `deltaCursor` , enabling delta polling via `mode: 'delta'` and `after: page.deltaCursor` for incremental updates after an initial load.

Poll only new trace completions after an initial page load, avoiding full rescans on each interval.

```typescript
const page = await client.queryTraces({ timeRange, pagination: {
page: 0, perPage: 100 } });

// On each poll interval:
const updates = await client.queryTraces({ timeRange, mode:
'delta', after: page.deltaCursor, limit: 100 });
```

— Named parameters and modes with a runnable polling example `@mastra/core@1.68.0`

05 **Durable agent snapshot persistence control** BREAKING 70

`shouldPersistSnapshot` predicate is added to createDurableAgent(), the `DurableAgent` constructor, and agent-level `durable` config to control which workflow states trigger storage writes. `running` checkpoints now only persist when `recovery.durableAgents: 'auto'` is set; callers using listActiveRuns(), recover(), or recoverActiveRuns() without that setting must opt back in via shouldPersistSnapshot: ({ workflowStatus }) => ['pending', 'paused', 'suspended', 'running'].includes(workflowStatus).

— Names the predicate, config key and required opt-in code but no runnable example

`@mastra/core@1.68.0`

06 **MCP v2 server rewrite with suspend/resume** BREAKING

70

`@mastra/mcp@2.0.0` is rebuilt on the MCP 2026-07-28 revision with first-class suspend/resume via context.suspend() and `context.resumeData`, signed self-contained `requestState`, and explicit `{ status: 'suspended', suspendPayload, resumeSchema }` vs `{ status: 'completed', output }` server responses. It is a major rewrite targeting the 2026-07-28 revision only: the legacy initialize handshake and session headers are removed, elicitation APIs are replaced by suspend/resume, and several server/client transport and protocol surfaces were removed.

— Names package version, protocol revision and removed APIs, no example

`@mastra/core@1.68.0`

07 **Programmatic skill validation functions** NEW

65

validateSkillContent(content, directoryName) validates skill markdown before writing files, returning a structured result (`result.errors` , `result.warnings`) instead of throwing. validateSkillMetadata({ metadata, directoryName }) validates an already-parsed frontmatter object against skill rules without re-parsing markdown.

Validate skill markdown written by a tool or shell command before persisting it, so CI fails fast on invalid content.

javascript

```
const result = validateSkillContent(skillMarkdown,
directoryName);
if (!result.valid) {
  throw new Error('Invalid skill:\n' + result.errors.join('\n'));
}
// result.warnings lists non-blocking issues, such as very long
instructions
```

Validate an already-parsed frontmatter object without re-parsing markdown, useful when the metadata is assembled programmatically.

```javascript
const result = validateSkillMetadata({ metadata, directoryName
});
if (!result.valid) {
  throw new Error('Invalid skill metadata:\n' +
result.errors.join('\n'));
}
```

— Two named functions with runnable code, no mechanism detail beyond that `product docs`

08 **Token-budgeted memory trimming** NEW 60

`messageHistory: { maxTokens, atMaxRemoveTokens? }` on Memory options trims conversation history to a token budget, persisting the trim boundary per thread.

Keep a Memory instance within a token budget so long conversations don't overflow the context window.

```typescript
const memory = new Memory({
  options: {
    messageHistory: { maxTokens: 8000, atMaxRemoveTokens: 2000 },
  },
});
```

— Named config keys with a runnable example, limited mechanism detail `@mastra/core@1.68.0`

09 **Root-span summary fields in trace queries** IMPROVED 60

queryTraces() results now include root-span summary fields — `name`, `entityId`, `parentSpanId`, `createdAt`, `metadata`, `inputPreview` — so UIs can render trace lists without fetching each full trace; also propagated to `@mastra/clickhouse`, `@mastra/duckdb`, and `@mastra/pg`.

— Fields and affected adapters named, no example `@mastra/core@1.68.0`

THINNER COVERAGE BELOW

10 **Server-side sorting for dataset and experiment list APIs** NEW 55

`orderBy: { field, direction }` is added to listDatasets(), listItems(), listExperiments(), and listExperimentResults() for server-side sorting; supported fields are `createdAt`, `updatedAt`, `name` (datasets), `createdAt`, `updatedAt` (items), `createdAt`, `status` (experiments), and `startedAt`, `createdAt` (experiment results).

– All four methods and their sortable fields named, no example `@mastra/core@1.68.0`

11 **Dataset snapshot validation utility** NEW 55

parseDatasetSnapshot() from `@mastra/core/datasets` validates portable dataset snapshot identities and detects artifact changes via an integrity digest, with a configurable `maxBytes` budget (default 4 MiB).

– Named function and default limit, no runnable example `@mastra/core@1.68.0`

12 **Bounded resumable storage pruning** NEW 55

Storage adapters gain a bounded, resumable prune() method accepting `maxBatches`, `maxRows`, `pauseMs`, and `signal` (AbortController) options.

– Named method and options, no example `@mastra/core@1.68.0`

13 **Agent Controller stream message event format change** BREAKING 50

Stream message events changed: a single full `message_start` is now emitted, followed by ID-addressed `message_update` deltas, and `message_end` contains only the message ID — consumers must store messages by ID and apply delta updates instead of receiving full messages.

– Names the three event types and required consumer change, no example `@mastra/core@1.68.0`

14 **Configurable searchable text extensions in filesystem** IMPROVED 45

`textExtensions` option on `MastraFilesystemOptions` (via `LocalFilesystem`) registers additional file extensions as searchable text, alongside built-in `.sas`, `.log`, and `.jsonl` support; unrecognized extensions are now reported as skipped in grep summaries.

– Named config option and behaviour, no example given `@mastra/core@1.68.0`

15 **Skip helper for unscoreable evals** NEW 45

notScorable() helper from `@mastra/core/evals` skips scorer steps when there is nothing to evaluate, omitting `verdict` from runEvals() results when no scorable gate or threshold applies.

– Named function and effect on runEvals(), no example `@mastra/core@1.68.0`

16 **New vector store integrations: Azure AI Search and Weaviate** NEW 45

`@mastra/azure-ai-search@0.1.0` adds a new `MastraVector` backend with auto-indexed filterable metadata for Memory and RAG workloads.

`@mastra/weaviate@0.1.0` adds another `MastraVector` backend with Mongo-style metadata filtering and deterministic vector ID mapping.

— Package names and headline capabilities given, no setup example `@mastra/core@1.68.0`

17 **Organization-level deploy and outage alerts** NEW 40

Mastra Platform adds organization-level Alerts that notify teams when deploys fail or running services stop, scoped to every project or to selected projects and environments, delivered via Slack, email, or webhooks.

— Describes scope and channels but no config surface or UI path `product docs`

18 **Agent peer advertisement refresh** NEW 30

agent.updateThreadPeerAdvertisement() refreshes advertised agent peer details (title, metadata) without reclaiming thread ownership.

— Bare method name and effect, no mechanism or example `@mastra/core@1.68.0`

└──▷ **BREAKING ON UPGRADE**

! Variables stored under any system-reserved name (`MASTRA_PROJECT_ID` , `MASTRA_ENVIRONMENT_ID` , `MASTRA_ENVIRONMENT_NAME` , `MASTRA_ENVIRONMENT_SLUG` , `MASTRA_PLATFORM_REGION` , `MASTRA_PLATFORM_ACCESS_TOKEN` , `MASTRA_PLATFORM_BUCKET_NAME` , `MASTRA_WORKERS`) are silently overridden at runtime by the platform-injected value — the stored value never reaches the running service.

! Agent Controller stream message events changed: a single full `message_start` is emitted, followed by ID-addressed `message_update` deltas; `message_end` now contains only the message ID — consumers must store messages by ID and apply delta updates.

! `@mastra/mcp@2.0.0` is a major rewrite targeting MCP 2026-07-28 only: the legacy initialize handshake and session headers are removed, elicitation APIs are replaced by suspend/resume (context.suspend() + `context.resumeData`), and several server/client transport and protocol surfaces were removed.

! Durable agents no longer persist `running` checkpoints by default; callers using listActiveRuns(), recover(), or recoverActiveRuns() without `recovery.durableAgents: 'auto'` must opt back in via shouldPersistSnapshot: ({ workflowStatus }) => [`pending` , `paused` , `suspended` , `running`].includes(workflowStatus) on createDurableAgent().

`@mastra/core@1.68.0` NOTES ↗ LATEST STABLE

Mastra adds MCP v2 server architecture, Azure AI Search and Weaviate vector stores, token-budgeted memory trimming, and richer trace pagination with delta polling.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @mastra/core@1.68.0 https://github.com/mastra-ai/mastra.git
# already have the repo? check out this version:
$ git checkout @mastra/core@1.68.0
```

└──▷ **USE IT**

Keep a Memory instance within a token budget so long conversations don't overflow the context window.

```typescript
const memory = new Memory({
  options: {
    messageHistory: { maxTokens: 8000, atMaxRemoveTokens: 2000 },
  },
});
```

Stream a generated thread title to an HTTP client immediately after a run completes, without polling.

```typescript
const memory = new Memory({
  options: {
    generateTitle: {
      emitEvent: true,
      minMessages: 2,
    },
  },
});

for await (const chunk of stream.fullStream) {
  if (chunk.type === 'data-thread-title') {
    console.log(chunk.data.threadId, chunk.data.title);
  }
}
```

Poll only new trace completions after an initial page load, avoiding full rescans on each interval.

```typescript
const page = await client.queryTraces({ timeRange, pagination: {
page: 0, perPage: 100 } });

// On each poll interval:
const updates = await client.queryTraces({ timeRange, mode: 'delta',
after: page.deltaCursor, limit: 100 });
```

Adds `textExtensions` option to `MastraFilesystemOptions` (via `LocalFilesystem`) to register additional file extensions as searchable text, alongside built-in `.sas`, `.log`, and `.jsonl` support; unrecognized extensions are now reported as skipped in grep summaries.

Adds `shouldPersistSnapshot` predicate to createDurableAgent(), `DurableAgent` constructor, and agent-level `durable` config to control which workflow states trigger storage writes; `running` checkpoints now only persist when `recovery.durableAgents: 'auto'` is set.

Adds `messageHistory: { maxTokens, atMaxRemoveTokens? }` to `Memory` options for token-budgeted conversation history trimming, persisting the trim boundary per thread.

Adds `generateTitle.emitEvent: true` to `Memory` options so run streams emit a `data-thread-title` chunk (with `threadId` and `title`) before the `finish` chunk, enabling HTTP/stream clients to receive generated thread titles without polling; also accepts `minMessages` and `model` sub-options.

Adds `orderBy: { field, direction }` parameter to listDatasets(), listItems(), listExperiments(), and listExperimentResults() for server-side sorting; supported fields are `createdAt`, `updatedAt`, `name` (datasets), `createdAt`, `updatedAt` (items), `createdAt`, `status` (experiments), `startedAt`, `createdAt` (experiment results).

+9 more

└──▷ **BREAKING ON UPGRADE**

! Agent Controller stream message events changed: a single full `message_start` is emitted, followed by ID-addressed `message_update` deltas; `message_end` now contains only the message ID — consumers must store messages by ID and apply delta updates.

! `@mastra/mcp@2.0.0` is a major rewrite targeting MCP 2026-07-28 only: the legacy initialize handshake and session headers are removed, elicitation APIs are replaced by suspend/resume (context.suspend() + `context.resumeData`), and several server/client transport and protocol surfaces were removed.

! Durable agents no longer persist `running` checkpoints by default; callers using listActiveRuns(), recover(), or recoverActiveRuns() without `recovery.durableAgents: 'auto'` must opt back in via shouldPersistSnapshot: ({ workflowStatus }) => [`pending` , `paused` , `suspended` , `running`].includes(workflowStatus) on createDurableAgent().

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ ai@7.0.112 NOTES ↗ RANK

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

> Vercel AI SDK added consumer cancellation reporting in UI message streams and telemetry support for evaluation runs.

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

01 **Telemetry support for experimental_evaluate** IMPROVED

40

Adds telemetry support to `experimental_evaluate`, enabling observability for evaluation runs.

– Names the API but not telemetry fields or setup. ai@7.0.111

02 **Consumer cancellation reporting in UI message streams** NEW

35

UI message stream end callbacks now report consumer cancellation events, enabling detection of client-side stream abandonment.

– Names the mechanism but no callback signature or API. ai@7.0.112

ai@7.0.112 NOTES ↗ LATEST STABLE

Vercel AI SDK 7.0.112 adds consumer cancellation reporting in UI message stream end callbacks.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch ai@7.0.112 https://github.com/vercel/ai.git
# already have the repo? check out this version:
$ git checkout ai@7.0.112
```

ai@7.0.111 NOTES ↗ STABLE

Vercel AI SDK 7.0.111 adds telemetry support to `experimental_evaluate`

└──▷ GET THIS VERSION ▾

```
$ git clone --branch ai@7.0.111 https://github.com/vercel/ai.git
# already have the repo? check out this version:
$ git checkout ai@7.0.111
```

WAS THIS USEFUL?  

Skyvern

Feedback ⓘ v1.0.54 NOTES ↗ RANK

Skyvern automates browser-based workflows with vision models and agents, completing form-driven tasks on sites it has not seen before.

> Skyvern's Search block gained configurable error codes, retry-on-validation-failure and a new Web Search block backed by Google and Exa (with automatic Google→Exa fallback), while Workflow Copilot picked up video-based authoring, draft editing and credential reuse; the release also adds workflow retry policies, Task V3 default rollout and security changes, user-selectable browser types, and gates passkey 2FA behind the enterprise plan.

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

01 Search block error codes, retries, and Google/Exa web search

NEW 90

The Search block gains `no_results_error_code`, which terminates the block and run with a configured code when a search returns no results (surfaced in the run's `errors` list and in `<block_label>_output.errors`), and `no_match_error_code`, which does the same when the Prompt reports no matching result and supports `continue_on_failure` so a downstream block can branch on it. The Prompt now retries once with validation feedback when its JSON response fails schema validation before failing the block, and using an array Data Schema lets a no-match answer return `[]` for downstream branching. A new Web Search workflow block is backed by Google and Exa as search providers, with automatic Google→Exa fallback.

Terminate a workflow run with a specific code when a Search block finds nothing, so a downstream block can detect the empty-result condition via `continue_on_failure`.

📍 In the Search block configuration, set No Results Error Code to a code such as `SEARCH_EMPTY`, then enable `continue_on_failure` on the block. The code will appear in the run's `errors` list and in `<block_label>_output.errors`, which a downstream block can inspect to branch accordingly.

Use an array Data Schema with a filter-style Prompt so a no-match returns `[]` rather than failing, enabling clean branching in downstream blocks.

📍 In the Search block, set the Prompt to your filter criteria and configure Data Schema as an array type. When the LLM finds no match it returns '[]', which downstream blocks can branch on instead of treating the result as a hard failure.

— Names exact config fields and includes runnable configuration steps. v1.0.54

02 **Workflow Copilot: video authoring, draft editing, credential reuse**

IMPROVED — 60

Workflow Copilot can now author workflows from demonstration videos and edit pending workflow drafts. It can also find and read the org's saved workflows, and at the credential checkpoint it now shows saved logins (not just create-new), reusing saved credentials and recovering the prior login selection.

— Names several concrete Copilot capabilities but no exact UI paths. v1.0.54

THINNER COVERAGE BELOW

03 **Task V3 default rollout, URL masking, and set selections**

IMPROVED — 55

New non-enterprise workflow task blocks now run on Task V3 by default. Signed URLs rendered into Task V3 goals, system guidance, or starting URLs are now hidden behind opaque tokens, and Task V3 can hold and report a set-valued selection.

— Names three concrete Task V3 changes but no config or endpoint given. v1.0.54

04 **Workflow retry policies and attempt-aware run views** NEW — 45

Adds workflow retry policy settings and attempt-aware run views, allowing a workflow run to be retried under the same run ID when it ends in a chosen status or error code.

— Explains mechanism but no config key or UI path given. v1.0.54

05 **Passkey 2FA gated to enterprise plan** BREAKING — 40

Gates passkey 2FA behind the enterprise plan, showing a visible book-a-demo upsell tile to users on other plans.

— Names the restriction and UI element but no migration detail. v1.0.54

06 **User-selectable browser type for runs** NEW — 35

Adds user-selectable browser type for workflows and runs, with backend support for choosing which browser type executes a run.

— Bare description with no interface details given. v1.0.54

07 **Intent-aware refinement in Record Browser** IMPROVED — 25

Adds intent-aware refinement of code-first recordings in the Record Browser feature.

— One-line description with no mechanism or usage detail. v1.0.54

08 **CDP-first screenshot experiment** NEW — 25

Adds a controlled CDP-first screenshot experiment.

— Bare mention of an experimental feature with no further detail. v1.0.54

v1.0.54 NOTES ↗ LATEST STABLE

Skyvern v1.0.54 adds Web Search blocks (Google + Exa), workflow retry policies, video-based Copilot authoring, and user-selectable browser types.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.0.54 https://github.com/Skyvern-AI/skyvern.git
# already have the repo? check out this version:
$ git checkout v1.0.54
```

Adds a Web Search workflow block backed by Google and Exa as search providers.

Adds workflow retry policy settings and attempt-aware run views, allowing a workflow run to be retried under the same run ID when it ends in a chosen status or error code.

Adds user-selectable browser type for workflows and runs (backend support).

Lets Workflow Copilot author workflows from demonstration videos.

Lets Copilot edit pending workflow drafts.

+9 more

WAS THIS USEFUL? 👍 👎

CopilotKit Feedback ⓘ v1.73.3 NOTES ↗ RANK

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

> CopilotKit's latest releases add a conversation-history trimming option, ease Express 5 compatibility, and simplify onboarding with a copyable hosted prompt link.

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

01 **messageFilter option for conversation history** NEW 48

Adds a `messageFilter` option to trim conversation history before it is sent to the model.

— Names the config option but not filter syntax or behavior. v1.73.1

02 **Express made optional peer dependency** IMPROVED 40

Makes Express an optional peer dependency so Express 5 apps can compile against the runtime without version conflicts.

— Explains the change and its purpose but no migration steps. v1.73.1

03 **Hosted prompt link in onboarding** IMPROVED 28

The onboarding flow now lets you copy a hosted prompt link instead of running the CLI command.

— Describes the change but no specifics on link format or CLI command. v1.73.3

v1.73.3 NOTES ↗ LATEST STABLE

Onboarding flow now lets you copy a hosted prompt link instead of running the CLI command.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.73.3
https://github.com/CopilotKit/CopilotKit.git
# already have the repo? check out this version:
$ git checkout v1.73.3
```

v1.73.1 NOTES ↗ STABLE

CopilotKit v1.73.1 adds a `messageFilter` for trimming conversation history and makes Express an optional peer dependency for Express 5 compatibility.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.73.1
https://github.com/CopilotKit/CopilotKit.git
# already have the repo? check out this version:
$ git checkout v1.73.1
```

Adds `messageFilter` option to trim conversation history before it is sent to the model.

Makes Express an optional peer dependency so Express 5 apps can compile against the runtime without version conflicts.

WAS THIS USEFUL? 👍 👎

Langflow

Feedback ⓘ v1.12.3 NOTES ↗ RANK

Langflow is a visual builder for agents and RAG workflows that exports each flow as an API or Python object for deployment.

> Langflow's v1.12.3 release adds assistant UX and streaming controls, alongside a breaking default change to local file access restrictions.

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

01 **Local file access restriction now defaults on** BREAKING

85

The `LANGFLOW_RESTRICT_LOCAL_FILE_ACCESS` environment variable now defaults to `true`, which will block local file access that previously worked without this flag being explicitly set.

— Exact env var and clear before/after behavior for migration v1.12.3

THINNER COVERAGE BELOW

02 **Slash command menu and step budget in Agentic Assistant** NEW

50

The Agentic Assistant gains a slash command menu and a 100-step budget limit.

— Names the feature and a numeric limit but no usage mechanics v1.12.3

03 **Graph state opt-out on AG-UI stream** IMPROVED

40

API callers can now opt out of graph state inclusion on the AG-UI stream.

— States capability but no parameter name or example given v1.12.3

└──▷ **BREAKING ON UPGRADE**

! The `LANGFLOW_RESTRICT_LOCAL_FILE_ACCESS` environment variable now defaults to `true`, which will block local file access that previously worked without this flag being explicitly set.

v1.12.3 NOTES ↗ LATEST STABLE

Langflow v1.12.3 adds a slash command menu in the assistant and lets API callers opt out of graph state on streams.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.12.3 https://github.com/langflow-ai/langflow.git
# already have the repo? check out this version:
$ git checkout v1.12.3
```

Adds a slash command menu and 100-step budget to the Agentic Assistant.

Lets API callers opt out of graph state inclusion on the AG-UI stream.

└─▷ **BREAKING ON UPGRADE**

! The `LANGFLOW_RESTRICT_LOCAL_FILE_ACCESS` environment variable now defaults to `true`, which will block local file access that previously worked without this flag being explicitly set.

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ python/v1.57.0 NOTES ↗

RANK

Strands Agents is AWS's open-source SDK for production agents in Python and TypeScript, with a model-driven loop, tool integration, and deployment targets.

> Strands Agents expanded its harness SDK heavily this window: a sandboxed programmatic tool caller (Monty), a full built-in shell/file toolset, MCP server integration, memory and Agent Skills systems, context/caching controls, background task execution, plus new vended tools (mcp_router, handoff_to_user) and Bedrock connectivity improvements across the Python and TypeScript SDKs.

└──▷ **WHAT SHIPPED · 16 FEATURES** most completely described first ? what's the number?

01 Shell and file built-in tools NEW

95

Adds `shell`, a stateless sandbox-routed command tool accepting `command` and optional `timeout` with no persisted working directory or environment across calls; `read`, which returns `cat -n -style` numbered lines with `offset` / `limit` (default limit 2000) and can return images (`png`, `jpg`, `jpeg`, `gif`, `webp`) and documents (`pdf`, `doc`, `docx`, `xls`, `xlsx`) as viewable media; `write`, which creates or overwrites a file at an absolute path; and `edit`, which replaces an exact string via `old_str` and fails on ambiguous matches. Any subset can be enabled via `builtin_tools` (e.g. read-only access), and all four route through the same sandbox seam supporting local, Docker, or SSH backends.

Restrict an agent to read-only file access by narrowing `builtin_tools` — useful when the agent should audit a project without making changes.

```yaml
builtin_tools:
  - read
```

Chain directory change and build in one `shell` call since each invocation starts a fresh shell with no persisted working directory.

```python
shell(command='cd /repo && cmake . && make -j4')
```

— Complete mechanism, config key, and runnable examples for all four tools. product docs

02 MCP server integration NEW

95

Adds `mcp_servers` (Python) / `mcpServers` (TypeScript) to create_harness()/createHarness(), accepting a path to a JSON file or an inline `{name: {...}}` mapping to connect Model Context Protocol servers and expose their tools. Per-server keys `"continue_on_error": false` make a server's startup failure fatal instead of silently yielding no tools, and `"prefix"` namespaces a server's tools to avoid clashes; a `--mcp-config` CLI flag supplies additional config files alongside automatic discovery of project MCP config.

Connect a filesystem MCP server inline when constructing a harness agent, avoiding the need for a separate config file.

python

```python
from strands_harness import create_harness

agent = create_harness(
    mcp_servers={
        "filesystem": {
            "command": "npx",
            "args": ["-y", "@modelcontextprotocol/server-
filesystem", "."]
        }
    }
)
```

Mark a critical MCP server so its failure aborts agent construction rather than silently dropping its tools.

json

```json
{
  "mcpServers": {
    "filesystem": {
      "command": "npx",
      "args": ["-y", "@modelcontextprotocol/server-filesystem",
"."],
      "continue_on_error": false
    },
    "search": {
      "command": "npx",
      "args": ["-y", "@modelcontextprotocol/server-search"],
      "prefix": "search"
    }
  }
}
```

— All named config keys and flag with runnable examples. product docs

03 Context management and conversation offloading NEW

95

Adds `context_manager` to create_harness() (`'auto'`, `'agentic'`, or `False` / `null` / `off`) and `contextManager` to createHarness() (`'auto'` / `'agentic'`) to control summarization and context offloading; adds `SlidingWindowConversationManager` (`strands.agent.conversation_manager` / `@strands-agents/sdk`) with `window_size` and `should_truncate_results` for direct SDK control. Offloading moves bulky tool results to storage, replacing them with a short preview plus a reference, persisting under the session directory when a session is active.

Use agentic context management to summarize older turns and offload large tool results automatically in a long-running task.

```python
from strands_harness import create_harness

agent = create_harness(
    context_manager='agentic'
)
```

Cap the conversation window and truncate oversized tool results when using the core SDK directly, without the Harness layer.

```python
from strands import Agent
from strands.agent.conversation_manager import
SlidingWindowConversationManager

conversation_manager = SlidingWindowConversationManager(
    window_size=20,
    should_truncate_results=True
)
agent = Agent(conversation_manager=conversation_manager)
```

— Named APIs across Python and TypeScript with runnable examples. product docs

04 Memory store configuration and lifecycle NEW

95

Exposes a `search_memory` tool for on-demand recall during any turn. `memory={"dir": ...}` relocates the file-based memory store (defaults to `./.agent/memory`); `memory={"stores": [...]}` swaps in a custom backend while preserving harness injection and the `search_memory` policy; `memory=False` / `memory: false` disables memory entirely. agent.shutdown() / agent.shutdown_async() (Python) and agent.shutdown() (TypeScript) flush pending

background memory extraction before a short-lived process exits, and scoped lifetimes via `async with create_harness() as agent` (Python) or await using agent = await createHarness() (TypeScript) flush automatically on block exit.

Store agent memory in a shared path (e.g. a Docker volume) so facts persist across container restarts and deployments.

python

```
from strands_harness import create_harness

agent = create_harness(
    memory={"dir": "/var/lib/agent/memory"}
)
```

Plug in a durable backend (e.g. Postgres) so recalled facts survive beyond the local filesystem, without losing harness injection defaults.

python

```
from strands_harness import create_harness
from my_stores import PostgresMemoryStore

agent = create_harness(
    memory={"stores":
[PostgresMemoryStore(dsn="postgres://user:pass@host/db")]}
)
```

Ensure pending memory extractions are flushed when running a short-lived script that invokes the agent once and exits.

python

```
import asyncio
from strands_harness import create_harness

async def main():
    async with create_harness() as agent:
        await agent.invoke_async("Summarise today's threat intel
report")
        # shutdown and memory flush happen automatically here

asyncio.run(main())
```

— Every memory config surface named with runnable examples. product docs

05 **Harness factory configuration surface** NEW 90

Adds `interventions` / `interventions` to gate tool calls behind approval or a policy; `plugins` / `plugins` for consumer plugins alongside built-ins; `builtin_plugins` / `builtinPlugins` to control built-in feature plugins (default `["todos", "environment"]`, or `[]` for none); `instructions` / `instructions` for a domain block appended after the harness contract (ignored when a full `system_prompt` is passed); `tools` / `tools` for consumer tools alongside built-ins; and exports `HARNESS_CONTRACT`, `build_system_prompt` / `buildSystemPrompt`, `resolve_memory`, and `resolve_interventions` (Python only) as composition building blocks. Unrecognized keywords are forwarded to the underlying `Agent` constructor, and in TypeScript `HarnessAgentOptions` extends `AgentConfig` (minus harness-managed fields) so fields like `retryStrategy` pass through verbatim.

Connect an MCP server and restrict built-in tools to read-only access when standing up a harness for a code-review agent.

```python
harness = create_harness(
    mcp_servers="./mcp_servers.json",
    builtin_tools=["read", "web_fetch"],
    effort="high",
)
```

Persist a long-running investigation session by id so it can be resumed across runs, storing artifacts in a custom directory.

```python
harness = create_harness(
    session={"id": "incident-2025-001", "dir":
"/var/agent/sessions"},
    memory={"dir": "/var/agent/memory"},
    interventions="approval",
)
```

Swap the memory backend to a custom store while keeping harness policy, and disable background tasks for a tightly scoped scan.

```python
harness = create_harness(
    memory={"stores": [my_custom_store]},
    background_tasks={},
    builtin_plugins=["todos"],
)
```

— Every new factory argument named with three runnable examples. product docs

06 **Background task execution controls** NEW 85

Adds a `backgroundTasks` / `background_tasks` option to createHarness()/create_harness() accepting `false` to disable background execution entirely, or a policy object such as `{ agentic: ['long_running_report'] }` to control which tools run concurrently, their concurrency limits, completion behavior, and timeouts; defaults to `{ agentic: ['*'] }`.

Restrict background execution to one slow tool instead of allowing any compatible tool to run in the background.

```javascript
import { createHarness } from '@strands-agents/harness'

const harness = await createHarness({
  backgroundTasks: { agentic: ['long_running_report'] },
})
```

Disable background tasks entirely when you need strictly sequential tool execution.

```javascript
import { createHarness } from '@strands-agents/harness'

const harness = await createHarness({ backgroundTasks: false })
```

— Runnable examples and documented default policy. product docs

07 **Prompt caching controls** NEW 85

Adds a `caching` parameter to create_harness()/createHarness() (`"auto"` / True on by default, False / None to disable) to control Harness-configured cache points on Amazon Bedrock and Anthropic direct providers, and a `CacheConfig` class (`strands.models.CacheConfig` ; TypeScript `BedrockModel` `cacheConfig` field) to configure prompt caching at the model level via `tools_ttl` (Python) / `ttl` (TypeScript).

Enable explicit prompt-caching TTL on a Bedrock model to cut cost on repeated long conversations with Claude.

```python
from strands import Agent
from strands.models import BedrockModel, CacheConfig

bedrock_model = BedrockModel(
    model_id='global.anthropic.claude-sonnet-5',
    cache_config=CacheConfig(tools_ttl=True),
)
agent = Agent(model=bedrock_model)
```

— CacheConfig fields named; one runnable example provided. product docs

08 **Agent Skills for on-demand instructions** NEW 85

Adds an `AgentSkills` plugin (Python: `from strands import AgentSkills`; TypeScript: `@strands-agents/sdk/vended-plugins/skills`) that loads reusable instruction packages on demand. The `skills` parameter to create_harness()/AgentSkills() accepts directory paths (e.g. `'./team-skills'`, `'./project-skills'`), defaulting to `./.agent/skills`; passing `skills=None` / `skills: null` (or `off` on the CLI) disables loading even if a skills directory exists. Each skill is a subdirectory with a `SKILL.md` in `agentskills.io` format — front-matter metadata plus instructions — loaded only when a task calls for it.

Wire shared team skills into a Strands agent so it loads task-specific instructions on demand without bloating the system prompt.

```python
from strands import Agent, AgentSkills

plugin = AgentSkills(skills=["./team-skills", "./project-skills"])
agent = Agent(plugins=[plugin])
```

— Config key and format named; one runnable example. product docs

09 **Programmatic tool caller with sandboxed execution** NEW 70

Adds the `programmatic_tool_caller` built-in tool, on by default and removable via `builtin_tools`, letting the model invoke multiple registered tools inside a short Python program to chain, loop, filter, and parallelize calls within a single turn. The program runs inside Monty, an isolated Python interpreter with no filesystem, network, environment, or process access besides the registered tool functions, under memory, time, and recursion limits with capped output; calls made from inside the program are

still subject to the agent's existing autonomous intervention policies (Cedar policies or natural-language risk policies), which can block a call with a catchable error.

— Mechanism and config key named; no runnable example given. product docs

10 **Bedrock model connectivity: timeout and endpoint routing** NEW

70

Adds a `requestTimeout` option to `BedrockModel` to control per-request timeout on Bedrock calls (Python and TypeScript), and support for selecting the `bedrock-runtime` endpoint from `bedrock_mantle_config`, enabling custom endpoint routing for Bedrock models.

Cap how long a single Bedrock API call can block before timing out — useful in latency-sensitive pipelines where a hung model call would stall the whole agent.

python

```python
from strands.models import BedrockModel
from strands import Agent

model = BedrockModel(requestTimeout=30)
agent = Agent(model=model)
agent("Summarize the findings from last week's pen-test report")
```

Set a per-request timeout on a Bedrock model to avoid hung calls in latency-sensitive pipelines.

typescript

```typescript
import { BedrockModel } from '@strands-agents/sdk'

const model = new BedrockModel({
  requestTimeout: 30000
})
```

— Named field and config with runnable examples in both SDKs. typescript/v1.19.0 python/v1.57.0

11 **Vended tools: mcp_router and handoff_to_user** NEW

60

Adds `mcp_router` as a new vended tool for routing MCP requests from Python agents, and `handoff_to_user` as a new vended tool to explicitly hand off agent execution to a human; `handoff_to_user` is ported to TypeScript in v1.19.0, matching the Python implementation.

Give a human-in-the-loop agent an explicit exit point — hand control back to a user mid-task instead of letting the model loop indefinitely.

```python
from strands_harness import create_harness
from strands_tools import handoff_to_user

agent = create_harness(tools=[handoff_to_user])
agent("Review the deployment plan and ask the operator to confirm
before proceeding")
```

Fan tool calls across multiple MCP servers from a single Python agent using the new router vended tool.

```python
from strands_harness import create_harness
from strands_tools import mcp_router

agent = create_harness(tools=[mcp_router])
agent("Search the internal knowledge base and then open a Jira
ticket with the results")
```

Signal a clean handoff to a human operator from within a Python agent tool chain.

```python
from strands import Agent
from strands_tools import handoff_to_user

agent = Agent(tools=[handoff_to_user])
agent('Triage this alert and hand off to an analyst if it looks
critical')
```

— Tool names given with runnable examples but thin on internals. `typescript/v1.19.0` `python/v1.57.0`

THINNER COVERAGE BELOW

12 **Non-clobbering seam for Anthropic-direct server-side tools**

IMPROVED 40

Adds a non-clobbering seam for Anthropic-direct server-side tools (e.g. web search) so native provider tools are no longer overwritten by agent-configured tools.

— Names the behavior fixed but no config or example. `typescript/v1.19.0`

13 **Harness SDK monorepo merge** IMPROVED 40

Merges the Strands harness into the SDK monorepo, exposing create_harness() (Python) and createHarness() (TypeScript) as the top-level entry points.

— Names entry points but no further mechanism or example. typescript/v1.19.0

14 **Bidirectional streaming refinements** IMPROVED 35

Distinguishes audio deltas from complete content in bidirectional streaming, enabling finer-grained audio event handling, and accepts shared content blocks as input in bidirectional streaming sessions.

— No example, flag, or config name given. python/v1.57.0

15 **Default model upgraded to Opus 5** IMPROVED 35

create_harness()/createHarness() now default to Opus 5 with high thinking enabled.

— States the new default only, no further detail. python/v1.57.0

16 **Snapshot session management for Graph and Swarm** NEW 25

Supports snapshot session management for Graph and Swarm multi-agent patterns in the Python SDK.

— Bare mention with no mechanism or example. python/v1.57.0

python/v1.57.0 NOTES ↗ LATEST STABLE

Strands Agents v1.57.0 adds mcp_router tool, snapshot sessions for Graph/Swarm, handoff_to_user tool, and requestTimeout on BedrockModel

└──▷ GET THIS VERSION ▾

```
$ git clone --branch python/v1.57.0 https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout python/v1.57.0
```

└──▷ USE IT

Give a human-in-the-loop agent an explicit exit point — hand control back to a user mid-task instead of letting the model loop indefinitely.

```python
from strands_harness import create_harness
from strands_tools import handoff_to_user

agent = create_harness(tools=[handoff_to_user])
agent("Review the deployment plan and ask the operator to confirm
before proceeding")
```

Cap how long a single Bedrock API call can block before timing out — useful in latency-sensitive pipelines where a hung model call would stall the whole agent.

```python
from strands.models import BedrockModel
from strands import Agent

model = BedrockModel(requestTimeout=30)
agent = Agent(model=model)
agent("Summarize the findings from last week's pen-test report")
```

Fan tool calls across multiple MCP servers from a single Python agent using the new router vended tool.

```python
from strands_harness import create_harness
from strands_tools import mcp_router

agent = create_harness(tools=[mcp_router])
agent("Search the internal knowledge base and then open a Jira
ticket with the results")
```

Adds `requestTimeout` option to `BedrockModel` to control per-request timeout on Bedrock calls.

Adds `mcp_router` as a new vended tool for routing MCP requests from Python agents.

Adds `handoff_to_user` as a new vended tool to explicitly hand off agent execution to a human.

Supports snapshot session management for `Graph` and `Swarm` multi-agent patterns in the Python SDK.

Selects the `bedrock-runtime` endpoint from `bedrock_mantle_config`, enabling custom endpoint routing for Bedrock models.

+3 more

typescript/v1.19.0 NOTES ↗ STABLE

Strands Agents: Ports the `handoff_to_user` vended tool to TypeScript, matching the existing Python implementation.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch typescript/v1.19.0 https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout typescript/v1.19.0
```

└──▷ **USE IT**

Set a per-request timeout on a Bedrock model to avoid hung calls in latency-sensitive pipelines.

typescript

```typescript
import { BedrockModel } from '@strands-agents/sdk'

const model = new BedrockModel({
  requestTimeout: 30000
})
```

Signal a clean handoff to a human operator from within a Python agent tool chain.

python

```python
from strands import Agent
from strands_tools import handoff_to_user

agent = Agent(tools=[handoff_to_user])
agent('Triage this alert and hand off to an analyst if it looks critical')
```

Merges the Strands harness into the SDK monorepo, exposing create_harness() (Python) and createHarness() (TypeScript) as the top-level entry points.

Adds support for selecting the `bedrock-runtime` endpoint from `bedrock_mantle_config` on Bedrock model providers.

Adds a non-clobbering seam for Anthropic-direct server-side tools (e.g. web search) so native provider tools are no longer overwritten by agent-configured tools.

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

Composio

Feedback ⓘ @composio/core@0.20.0 NOTES ↗ RANK

Composio connects AI agents to external tools and services with managed authentication and integrations.

> Composio's SDK (@composio/core 0.20.0) gained a much larger client surface — webhook, logs, connected-account, toolkit and keyring APIs — plus stricter API-key scoping and optimistic-concurrency controls on session updates, while separate docs updates added session sandboxes, an MCP endpoint on sessions, TypeSafe tool selection, Cloudflare Workers deployment and Gmail trigger webhooks.

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

01 Stricter `apiKey` validation and null-disables-auth behavior

90

BREAKING

Passing `apiKey: null` to `ComposioConfig` now explicitly disables project-key authentication (including `COMPOSIO_API_KEY` and `~/.composio/user_data.json` fallbacks), throwing unless another credential (`userApiKey`, `orgApiKey`, or `x-user-api-key` in `defaultHeaders`) is configured — pass `undefined` (or omit `apiKey`) to keep the previous fallback behavior. An `x-api-key` entry in `defaultHeaders` is now rejected when `apiKey: null`, including on createSession() clones. Passing an invalid user API key (`uak_...`) as the project `apiKey` now throws `ComposioAPIKeyKindError` with a redacted diagnostic instead of silently sending a 401-bound request.

— Full before/after behavior with exact fields and error types named.

@composio/core@0.20.0

02 Optimistic concurrency for session updates BREAKING

90

Adds `expectedConfigVersion` parameter to session.update(): omitting it sends the last observed `configVersion` as a 409-safe precondition, while passing `expectedConfigVersion: false` gives last-writer-wins behavior. session.update() now sends `expected_config_version` by default and never retries on conflict — concurrent updates surface as `ComposioSessionConfigConflictError` (HTTP 409) instead of silently overwriting.

— Full mechanism, default behavior and the error/status code it now raises.

@composio/core@0.20.0

03 **User- and org-scoped API key config** NEW 87

Adds `userApiKey` and `orgApiKey` to `ComposioConfig`, falling back to `COMPOSIO_USER_API_KEY` and `COMPOSIO_ORG_API_KEY` environment variables, for user- and org-scoped API operations without mixing in the project key. Also adds `orgId` and `projectId` to `ComposioConfig`, sent as `x-org-id` / `x-project-id` headers on every request and included in session MCP config for consumer-project scoping.

Initialize the SDK with org-scoped credentials and a structured logger (e.g. pino) so all SDK output is routed through your existing log pipeline at debug level.

javascript

```
import { Composio } from '@composio/core';
import pino from 'pino';

const composio = new Composio({
  apiKey: process.env.COMPOSIO_PROJECT_API_KEY,
  userApiKey: process.env.COMPOSIO_USER_API_KEY,
  orgId: process.env.COMPOSIO_ORG_ID,
  projectId: process.env.COMPOSIO_PROJECT_ID,
  logger: pino(),
  logLevel: 'debug'
});
```

— Names exact config fields, env vars, headers, with a runnable example.

`@composio/core@0.20.0`

04 **Structured logging via `logger` and `logLevel` config** NEW 85

Adds `logger` and `logLevel` options to new Composio({...}): `logger` accepts any `{ error, warn, info, debug }` sink (e.g. `console`, pino, winston) and `logLevel` accepts `'silent' | 'error' | 'warn' | 'info' | 'debug'`, overriding `COMPOSIO_LOG_LEVEL`.

Initialize the SDK with org-scoped credentials and a structured logger (e.g. pino) so all SDK output is routed through your existing log pipeline at debug level.

```javascript
import { Composio } from '@composio/core';
import pino from 'pino';

const composio = new Composio({
  apiKey: process.env.COMPOSIO_PROJECT_API_KEY,
  userApiKey: process.env.COMPOSIO_USER_API_KEY,
  orgId: process.env.COMPOSIO_ORG_ID,
  projectId: process.env.COMPOSIO_PROJECT_ID,
  logger: pino(),
  logLevel: 'debug'
});
```

— Exact option names, accepted values and an env override, with a code sample.

`@composio/core@0.20.0`

05 **Webhook subscriptions and endpoints management API** NEW

85

Adds `composio.webhooks.subscriptions` with `list`, `get`, `set`, `update`, `delete`, `rotateSecret`, and `listEventTypes` methods, and `composio.webhooks.endpoints` with `list`, `get`, `create`, `replace`, and `update` methods. composio.triggers.setWebhookSubscription() now delegates to webhooks.subscriptions.set() and rejects malformed API responses with a `ValidationError` instead of reading them leniently.

— Enumerates every method on both new API groups plus a related behavior change.

`@composio/core@0.20.0`

06 **New toolkit retrieval, scope and custom-toolkit APIs** NEW

83

Adds composio.toolkits.getMany(slugs) and composio.toolkits.changelog() for bulk toolkit retrieval and changelog access. Adds composio.toolkits.recommendScopes(toolkitSlug, { tools, ... }) and composio.toolkits.listGrantContexts(toolkitSlug) for OAuth scope recommendations (API-marked beta). Adds `composio.experimental.customToolkits` with `upsert`, `sync`, and `delete` for project-owned custom toolkits (in pilot, shape may change).

— Enumerates every new toolkits method with signatures and stability status.

`@composio/core@0.20.0`

07 **Session policy null-clearing and config history** NEW

83

Session policy blocks (`toolkits`, `tools`, `tags`, `authConfigs`, `connectedAccounts`, `preload`, `manageConnections`, `multiAccount`, `search`,

`execute`, `experimental`) now accept `null` to remove stored overrides, with `manageConnections.callbackUrl: null` and `multiAccount.maxAccountsPerToolkit: null` removing individual sub-fields. Also adds session.listConfigHistory() on session objects.

— Enumerates every policy block and sub-field that gained null support.

`@composio/core@0.20.0`

---

08 **Scoped credential export in session MCP config** IMPROVED

76

Session MCP config now exports credentials from the auth context the session was actually made with (`x-api-key` for project key, or `x-user-api-key` for user key), and throws `ComposioMCPDestinationError` when the MCP URL origin differs from the configured API base URL.

— Explains the exact scoping mechanism and the new guard error. `@composio/core@0.20.0`

---

09 **New OAuth completion, keyring and auth-scheme APIs** NEW

70

Adds composio.connectedAccounts.completeAuth({ userId, sessionUri }) to complete a deferred OAuth connection after callback verification, composio.keyring.listTransferKeys() to retrieve the public JWKs of an organization's customer-managed keyring, and `CIMD_OAUTH` to `AuthSchemeTypes`.

— Names all three additions with exact signatures and purpose. `@composio/core@0.20.0`

---

10 **Connected account revocation method** NEW

65

Adds composio.connectedAccounts.revoke(), surfacing API 400/409 responses as `ComposioConnectedAccountRevocationNotSupportedError` / `ComposioConnectedAccountNotRevokableError`.

— Names the method and both typed errors it can raise. `@composio/core@0.20.0`

THINNER COVERAGE BELOW

---

11 **Gmail trigger webhooks with signature verification** NEW

55

Adds Gmail trigger webhook support, including signature verification and per-connected-user routing, enabling agents to run summarization tasks in response to incoming email events.

— Names Gmail, signature verification and routing behavior. `product docs`

---

12 **`connectedAccounts.refresh()` drops `validateCredentials`**

55

BREAKING

connectedAccounts.refresh() no longer sends `validateCredentials` since the upstream endpoint removed it, and now logs a warning when `validateCredentials` is set.

— States the removed field and the new warning behavior. `@composio/core@0.20.0`

13 **Tool-execution log retrieval API** NEW 54

Adds composio.logs.search() and composio.logs.get() for retrieving tool-execution logs.

— Names both methods but no further mechanism given. `@composio/core@0.20.0`

14 **Deploy Composio + OpenAI agents on Cloudflare Workers** NEW 48

Enables deploying authenticated Composio + OpenAI agent endpoints on Cloudflare Workers, written in TypeScript.

— Names the deployment target and stack, a clear starting point. `product docs`

15 **MCP endpoint exposed on Composio sessions** NEW 44

Composio sessions now expose an MCP (Model Context Protocol) endpoint, letting CrewAI and LangChain agents read available tools over MCP directly from a session.

— Names MCP, CrewAI, LangChain but no endpoint path or code. `product docs`

16 **Experimental usage summary and breakdown APIs** NEW 43

Adds composio.experimental.usage.summary() and composio.experimental.usage.breakdown() (experimental, shape may change).

— Names both methods but gives no further detail beyond experimental status. `@composio/core@0.20.0`

17 **Session sandboxes for file uploads and task execution** NEW 38

Composio sessions now support sandbox execution: agents can upload data files and execute tasks within a session sandbox, producing a downloadable report as output.

— Describes the capability in prose without naming an API surface. `product docs`

18 **TypeSafe tool selection via Jev** NEW 38

Adds TypeSafe tool selection via Jev, enabling partial call completion with known arguments before execution through Composio.

— Names Jev and the mechanism but gives no usage detail. `product docs`

19 **Combine remote and local tools in one agent run** NEW 28

Composio now supports combining remote Composio tools with local custom tools within a single agent run.

— Purely descriptive, no API or config named. `product docs`

└──▷ **BREAKING ON UPGRADE**

! Passing `apiKey: null` (e.g. `apiKey: process.env.COMPOSIO_API_KEY ?? null` when the variable is unset) no longer falls back to `COMPOSIO_API_KEY` or `~/.composio/user_data.json` — it now throws unless another credential (`userApiKey` , `orgApiKey` , or `x-user-api-key` in `defaultHeaders`) is configured. Pass `undefined` (or omit `apiKey`) to keep the previous fallback behavior.

! With `apiKey: null` , an `x-api-key` entry in `defaultHeaders` is now rejected, including on createSession() clones — pass the project key as `apiKey` instead.

! Providing a Composio user API key (`uak_...`) as the project `apiKey` now throws `ComposioAPIKeyKindError` instead of proceeding and failing with 401.

! composio.triggers.setWebhookSubscription() now delegates to webhooks.subscriptions.set() and rejects malformed API responses with a `ValidationError` instead of reading them leniently.

! connectedAccounts.refresh() no longer sends `validateCredentials` (the upstream endpoint removed it) and logs a warning when `validateCredentials` is set.

! session.update() now sends `expected_config_version` by default and never retries on conflict — concurrent updates surface as `ComposioSessionConfigConflictError` (HTTP 409) instead of silently overwriting.

`@composio/core@0.20.0` NOTES ↗ **LATEST** STABLE

Composio SDK gains `userApiKey` / `orgApiKey` config, a broad new client surface, structured logging, and safer MCP session credential scoping.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @composio/core@0.20.0
https://github.com/ComposioHQ/composio.git
# already have the repo? check out this version:
$ git checkout @composio/core@0.20.0
```

└──▷ **USE IT**

Initialize the SDK with org-scoped credentials and a structured logger (e.g. pino) so all SDK output is routed through your existing log pipeline at debug level.

```javascript
import { Composio } from '@composio/core';
import pino from 'pino';

const composio = new Composio({
  apiKey: process.env.COMPOSIO_PROJECT_API_KEY,
  userApiKey: process.env.COMPOSIO_USER_API_KEY,
  orgId: process.env.COMPOSIO_ORG_ID,
  projectId: process.env.COMPOSIO_PROJECT_ID,
  logger: pino(),
  logLevel: 'debug'
});
```

Adds `userApiKey` and `orgApiKey` to `ComposioConfig`, falling back to `COMPOSIO_USER_API_KEY` and `COMPOSIO_ORG_API_KEY` environment variables, for user- and org-scoped API operations without mixing in the project key.

Adds `orgId` and `projectId` to `ComposioConfig`, sent as `x-org-id` / `x-project-id` headers on every request and included in session MCP config for consumer-project scoping.

Adds `logger` and `logLevel` options to new Composio({...}): `logger` accepts any `{ error, warn, info, debug }` sink (e.g. `console`, pino, winston) and `logLevel` accepts `'silent' | 'error' | 'warn' | 'info' | 'debug'`, overriding `COMPOSIO_LOG_LEVEL`.

Passing `apiKey: null` to `ComposioConfig` now explicitly disables project-key authentication (including `COMPOSIO_API_KEY` and `~/.composio/user_data.json` fallbacks), enabling user-only auth via `userApiKey` or `orgApiKey`.

Passing an invalid user API key (`uak_...`) as the project `apiKey` now throws `ComposioAPIKeyKindError` with a redacted diagnostic instead of silently sending a 401-bound request.

+14 more

└─▷ **BREAKING ON UPGRADE**

! Passing `apiKey: null` (e.g. `apiKey: process.env.COMPOSIO_API_KEY ?? null` when the variable is unset) no longer falls back to `COMPOSIO_API_KEY` or `~/.composio/user_data.json` — it now throws unless another credential (`userApiKey`, `orgApiKey`, or `x-user-api-key` in `defaultHeaders`) is configured. Pass `undefined` (or omit `apiKey`) to keep the previous fallback behavior.

! With `apiKey: null`, an `x-api-key` entry in `defaultHeaders` is now rejected, including on createSession() clones — pass the project key as `apiKey` instead.

- ! Providing a Composio user API key (`uak_...`) as the project `apiKey` now throws `ComposioAPIKeyKindError` instead of proceeding and failing with 401.
- ! composio.triggers.setWebhookSubscription() now delegates to webhooks.subscriptions.set() and rejects malformed API responses with a `ValidationError` instead of reading them leniently.
- ! connectedAccounts.refresh() no longer sends `validateCredentials` (the upstream endpoint removed it) and logs a warning when `validateCredentials` is set.
- ! session.update() now sends `expected_config_version` by default and never retries on conflict — concurrent updates surface as `ComposioSessionConfigConflictError` (HTTP 409) instead of silently overwriting.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Google Gemini Enterprise Agent Platform

Feedback ⓘ changelog · Sep 23, 2026 CHANGELOG ↗ RANK

Gemini Enterprise Agent Platform provides Google's managed model training, inference, evaluation, and agent deployment services, formerly offered under Vertex AI.

> Gemini Enterprise Agent Platform's biggest change this window is a breaking SDK split that moves the agent surface into a new `google-cloud-agentplatform` package, alongside an overhauled interactive HTML security report, expanded CodeMender language coverage, new latency telemetry, and the addition of Claude Opus 5.5.

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

01 Interactive HTML security report for CodeMender IMPROVED

91

`cm report --format html` now produces an interactive dashboard with severity metric cards, syntax-highlighted code snippets, line numbers, and an inline patch diff viewer. A new `--open` (`-o`) flag launches the report directly in the default browser after generation.

Open an interactive HTML security report immediately after scanning, to triage findings with the inline patch diff viewer without manually locating the output file.

```
$ cm report --format html --open
```

— Names exact flags and shows a runnable command. changelog-20260923-eff3be8c

02 Expanded language coverage and discovery config for CodeMender

NEW

83

CodeMender's default discovery configuration and initialization templates now add out-of-the-box vulnerability scanning support for C# (`.cs`), Rust (`.rs`), Kotlin (`.kt`, `.kts`), Ruby (`.rb`), and PHP (`.php`). Scan scope can be tuned via `scan_config.exclude_dirs` in `.codemender/config.yaml` to skip build, vendor, node_modules, .gradle, dist, and .git directories across all `cm scan` runs.

Exclude build and vendor directories from all CodeMender scans by setting exclusion rules once in config, so every `cm scan` run skips them automatically.

```yaml
scan_config:
  exclude_dirs:
    - build
    - vendor
    - .git
```

Restrict CodeMender scans to exclude dependency and build directories to avoid noise in large monorepos.

```yaml
scan_config:
  exclude_dirs:
    - node_modules
    - .gradle
    - dist
    - vendor
```

— Lists languages, extensions, and a working config example. changelog-20260923-eff3be8c

THINNER COVERAGE BELOW

03 **Per-turn latency breakdowns in session exports** NEW

28

Session exports now include per-turn latency breakdowns that distinguish model inference wait time from local tool execution time.

— No field name, endpoint, or example given. changelog-20260923-eff3be8c

04 **Claude Opus 5.5 added to the platform** NEW

20

Anthropic's Claude Opus 5.5 is now available on Gemini Enterprise Agent Platform.

— Model name only, no usage or configuration detail. changelog-20260923-8a4e44d7 changelog-20260923-eff3be8c

└──▷ **BREAKING ON UPGRADE**

! Agent Platform SDK for Python 2.0.1 (`google-cloud-agentplatform`) decouples the agent surface from `google-cloud-aiplatform` into a dedicated package with restructured namespaces — existing code importing from `google-cloud-aiplatform` agent modules requires migration per the Agent Platform SDK for Python version 2.0.1 migration guide.

! The agent surface is decoupled from `google-cloud-aiplatform` into the new `google-cloud-agentplatform` package in SDK 2.0.1; users must migrate namespaces per the Agent Platform SDK for Python version 2.0.1 migration guide.

changelog · Sep 23, 2026 CHANGELOG ↗ LATEST STABLE

Gemini Enterprise Agent Platform: Overhauls `cm report --format html` to produce an interactive dashboard with severity metric cards, syntax-highlighted code snippets, line numbers, and an inline patch diff viewer; adds `--open` (`-o`) flag to launch the report in the default browser.

└──▷ **TRY IT**

Open an interactive HTML security report immediately after scanning, to triage findings with the inline patch diff viewer without manually locating the output file.

```
$ cm report --format html --open
```

Exclude build and vendor directories from all CodeMender scans by setting exclusion rules once in config, so every `cm scan` run skips them automatically.

```yaml
scan_config:
  exclude_dirs:
    - build
    - vendor
    - .git
```

Adds out-of-the-box vulnerability scanning support for C# (`.cs`), Rust (`.rs`), Kotlin (`.kt`, `.kts`), Ruby (`.rb`), and PHP (`.php`) to CodeMender's default discovery configuration and initialization templates.

Adds per-turn latency breakdowns to session exports, distinguishing model inference wait time from local tool execution time.

Adds Anthropic's Claude Opus 5.5 to the platform.

└──▷ **BREAKING ON UPGRADE**

! Agent Platform SDK for Python 2.0.1 (`google-cloud-agentplatform`) decouples the agent surface from `google-cloud-aiplatform` into a dedicated package with restructured namespaces — existing code importing from `google-cloud-aiplatform` agent modules requires migration per the Agent Platform SDK for Python version 2.0.1 migration guide.

changelog · Sep 23, 2026 CHANGELOG ↗ STABLE

Gemini Enterprise Agent Platform: Anthropic's Claude Opus 5.5 is now available on Gemini Enterprise Agent Platform.

└──▷ **USE IT**

Restrict CodeMender scans to exclude dependency and build directories to avoid noise in large monorepos.

.codemender/config.yaml

```
scan_config:
  exclude_dirs:
    - node_modules
    - .gradle
    - dist
    - vendor
```

Upgrade to the new decoupled Agent Platform Python SDK and verify the migration away from google-cloud-aiplatform.

```
$ pip install google-cloud-agentplatform==2.0.1
```

└──▷ **BREAKING ON UPGRADE**

! The agent surface is decoupled from `google-cloud-aiplatform` into the new `google-cloud-agentplatform` package in SDK 2.0.1; users must migrate namespaces per the Agent Platform SDK for Python version 2.0.1 migration guide.

WAS THIS USEFUL? 👍 👎

Perplexity API

Feedback ⓘ snapshot-20260923 NOTES ↗ RANK

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

> Perplexity expanded its Agent API with multi-provider orchestration, an AG2 integration guide, four new model options, and new cookbook examples for data analytics and regression testing.

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

01 **New Agent API model options** NEW 63

The Agent API adds four new supported models: `openai/gpt-6-sol`, `openai/gpt-6-luna`, `anthropic/claude-opus-5-5`, and `xai/grok-4.7`.

– Exact model identifiers given, ready to use snapshot-20260923

02 **Multi-provider orchestration in Agent API** NEW 62

The Agent API now supports routing between OpenAI, Anthropic, Google, and xAI models, with fallback chains and side-by-side provider comparison.

– Describes mechanism but no config keys or endpoints product docs

THINNER COVERAGE BELOW

03 **AG2 integration guide for Agent API** NEW 50

A new integration guide shows how AG2's Perplexity tools use the Search API and Sonar Chat Completions during migration to the Agent API.

– Names APIs involved but no runnable steps given product docs

04 **New Agent API cookbook examples** NEW 47

Adds a Data Analytics Agent cookbook example using Custom Connectors to inspect data and generate charts via the Agent API, and a regression-testing cookbook example for catching unsafe retry advice and unsupported answers before switching models.

– Names two cookbook use cases without code detail product docs

snapshot-20260923 NOTES ↗ LATEST STABLE

Perplexity Agent API gains four new model options: GPT-6 Sol, GPT-6 Luna, Claude Opus 5.5, and Grok 4.7.

Adds `openai/gpt-6-sol` as a supported model in the Agent API.

Adds `openai/gpt-6-luna` as a supported model in the Agent API.

> Adds `anthropic/claude-opus-5-5` as a supported model in the Agent API.
>
> Adds `xai/grok-4.7` as a supported model in the Agent API.

WAS THIS USEFUL?  

Together AI

Feedback ⓘ changelog · Sep 23, 2026 CHANGELOG ↗

RANK

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

> Together AI's headline addition is traffic rollout orchestration for endpoints (canary, blue-green, rolling strategies) via a new CLI and SDK surface, alongside HIPAA-attested deployment placement, a new TTS autoscaling metric, idle-timeout auto-shutdown, dedicated-container support for Flux2 image generation and Wan 2.1 video generation, and breaking changes to management API validation and the console's deployment-profile UI.

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

01 **Traffic rollout orchestration for endpoints** NEW 88

The new `tg beta endpoints rollout` CLI command and `client.beta.endpoints.rollouts` Python SDK surface let you shift live traffic between deployments under the same endpoint URL using canary, blue-green, or rolling strategies. Rollouts can be paused, resumed, promoted, or cancelled mid-flight, with an optional automatic pause gated on live metrics if the new deployment regresses.

— Names CLI command, SDK path, and strategies but no runnable example.

changelog-20260923-d4a5db3c

02 **HIPAA-attested cluster placement for deployments** NEW 76

The `--placement.hipaa` CLI flag and `compliance_policy` Python SDK field restrict a dedicated deployment to HIPAA-attested clusters, e.g. `tg endpoints create --model meta-llama/Llama-3-8b-hf --placement.hipaa`.

Restrict a new dedicated deployment to HIPAA-attested clusters to meet compliance requirements.

```
$ tg endpoints create --model meta-llama/Llama-3-8b-hf --placement.hipaa
```

— Names both the flag and SDK field with a runnable command. changelog-20260923-d4a5db3c

03 **Idle timeout auto-shutdown for dedicated endpoints** NEW 75

Dedicated deployments can be configured to automatically shut down and stop billing after a period of inactivity, using `tg endpoints update my-endpoint --inactive-`

`timeout 30` to set a 30-minute idle timeout when no inference traffic arrives.

Automatically shut down an idle dedicated deployment to stop billing when no inference traffic arrives for 30 minutes.

```
$ tg endpoints update my-endpoint --inactive-timeout 30
```

— Runnable command with exact flag and value shown. changelog-20260923-d4a5db3c

04 **WebSocket session autoscaling metric for TTS** NEW 65

Dedicated TTS endpoints can now autoscale using `active_sessions`, a new metric that targets the average number of concurrent open client WebSocket sessions per replica (`AVERAGE_VALUE` only).

— Names metric and scope but no config example given. changelog-20260923-d4a5db3c

05 **Management API rejects unknown request fields** BREAKING 62

The management API now rejects unknown JSON request-body fields and unknown query parameters with an HTTP error naming the offending field, instead of silently ignoring them; clients sending retired or misspelled keys, e.g. `inactive_timeout` instead of `inactiveTimeout`, will now receive errors.

— Names the exact field mismatch example but no endpoint path.

changelog-20260923-d4a5db3c

06 **Deployment profiles replace manual hardware controls** BREAKING 60

The dedicated endpoint create form in the console now uses Deployment profiles choice cards, deriving hardware, quantization, and decoding options from each profile's certified config. This removes the separate Quantization, Hardware, LoRA, and speculative-decoding controls that previously let users configure these independently.

— Describes UI change and removed controls but only a UI path.

changelog-20260923-d4a5db3c

THINNER COVERAGE BELOW

07 **New model types deployable via dedicated containers** NEW 52

Dedicated containers now support deploying a Flux2 image generation model and a Wan 2.1 multi-GPU video generation model on Together's managed GPU infrastructure, as well as serving a custom model behind an OpenAI-compatible endpoint for drop-in compatibility with code written against the OpenAI SDK.

— Names three new deployment targets but no commands or config. snapshot-20260923

└──▷ **BREAKING ON UPGRADE**

! The management API now rejects unknown JSON request-body fields and unknown query parameters with an HTTP error; clients that send retired or misspelled keys (e.g. `inactive_timeout` instead of `inactiveTimeout`) will receive errors instead of being silently ignored.

! The dedicated endpoint create form removes separate Quantization, Hardware, LoRA, and speculative-decoding controls; hardware and decoding options are now fixed by the selected Deployment profile.

changelog · Sep 23, 2026 CHANGELOG ↗ LATEST STABLE

Together AI: Adds `tg beta endpoints rollout` CLI command and `client.beta.endpoints.rollouts` Python SDK surface to shift live traffic between deployments under the same endpoint URL using canary, blue-green, or rolling strategies.

└──▷ **TRY IT**

Automatically shut down an idle dedicated deployment to stop billing when no inference traffic arrives for 30 minutes.

```
$ tg endpoints update my-endpoint --inactive-timeout 30
```

Restrict a new dedicated deployment to HIPAA-attested clusters to meet compliance requirements.

```
$ tg endpoints create --model meta-llama/Llama-3-8b-hf --
placement.hipaa
```

Adds `--placement.hipaa` CLI flag and `compliance_policy` Python SDK field to restrict a dedicated deployment to HIPAA-attested clusters.

Adds `active_sessions` as an autoscaling metric for TTS dedicated endpoints, targeting concurrent open client WebSocket sessions per replica (`AVERAGE_VALUE` only).

Management API now rejects unknown JSON request-body fields and unknown query parameters with an HTTP error naming the offending field instead of silently ignoring them.

Dedicated endpoint create form in the console now uses Deployment profiles choice cards, with hardware, quantization, and decoding options derived from each profile's certified config.

Rollouts can be paused, resumed, promoted, or cancelled mid-flight, with optional automatic pause gated on live metrics if the new deployment regresses.

└──▷ **BREAKING ON UPGRADE**

! The management API now rejects unknown JSON request-body fields and unknown query parameters with an HTTP error; clients that send retired or misspelled keys (e.g. `inactive_timeout` instead of `inactiveTimeout`) will receive errors instead of being silently ignored.

! The dedicated endpoint create form removes separate Quantization, Hardware, LoRA, and speculative-decoding controls; hardware and decoding options are now fixed by the selected Deployment profile.

snapshot-20260923 NOTES ↗ STABLE

Together AI adds Flux2 image gen, Wan 2.1 video gen, and OpenAI-compatible custom endpoints via dedicated containers.

Deploy a Flux2 image generation model on Together's managed GPU infrastructure using dedicated containers.

Deploy a Wan 2.1 multi-GPU video generation model on Together's managed GPU infrastructure using dedicated containers.

Serve a custom model behind an OpenAI-compatible endpoint using dedicated containers, enabling drop-in compatibility for code written against the OpenAI SDK.

WAS THIS USEFUL? 👍 👎

FlashAttention

Feedback ⓘ fa4-v4.0.0.beta32 NOTES ↗ RANK

FlashAttention implements fast, memory-efficient exact attention kernels used by most current transformer training and inference stacks.

> FlashAttention's fa4-v4.0.0.beta32 build extends sparse MLA support on SM100 (Blackwell) GPUs with a full backward pass and a wider Q-head range, alongside a new ROCm build option.

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

01 **Sparse MLA backward pass on SM100** NEW 65

Adds a sparse MLA backward pass on SM100 (Blackwell) GPUs with in-kernel P recomputation and token-chunked backward, enabling memory-efficient training on these GPUs.

– Mechanism named but no command or config to invoke it fa4-v4.0.0.beta32

02 **Sparse MLA Q-head range expanded on SM100** IMPROVED 65

Extends sparse MLA on SM100 (Blackwell) to support 1–128 Q heads via in-kernel TMA padding, up from a narrower fixed range.

– Clear before/after and mechanism, but no usage steps given fa4-v4.0.0.beta32

03 **ROCm system aiter build flag** NEW 65

Adds the `FLASH_ATTENTION_USE_SYSTEM_AITER` environment variable flag for ROCm builds, letting them use a system-installed aiter package instead of the bundled submodule.

– Names exact env var but no example of setting it fa4-v4.0.0.beta32

fa4-v4.0.0.beta32 NOTES ↗ PRE-RELEASE

FlashAttention fa4-v4.0.0.beta32 adds sparse MLA backward pass, ROCm system aiter flag, and broader SM100 Q-head support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch fa4-v4.0.0.beta32 https://github.com/Dao-AILab/flash-attention.git
# already have the repo? check out this version:
$ git checkout fa4-v4.0.0.beta32
```

> Adds `FLASH_ATTENTION_USE_SYSTEM_AITER` environment variable flag for ROCm builds to use a system-installed aiter package instead of the bundled submodule.
>
> Extends sparse MLA on SM100 (Blackwell) to support 1–128 Q heads via in-kernel TMA padding, up from a narrower fixed range.
>
> Adds sparse MLA backward pass on SM100 with in-kernel P recomputation and token-chunked backward, enabling memory-efficient training on Blackwell GPUs.

WAS THIS USEFUL? 👍 👎

SkyPilot

Feedback ⓘ v0.14.0rc1 NOTES ↗ RANK

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

> SkyPilot v0.14.0rc1 adds multi-server local API isolation, per-node Slurm job attribution and dashboard visibility, GCS federated credentials, and Kueue-aware provisioning timeouts, while dropping conda from default Kubernetes images as a breaking change.

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

01 **Per-node Slurm job attribution via SlurmClient.get_all_jobs_info()**

NEW 70

Adds SlurmClient.get_all_jobs_info() for per-node job attribution across Slurm clusters, letting jobs be attributed to specific nodes rather than only the cluster level.

– Named method but no usage example given v0.14.0rc1

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

65

Excludes Kueue admission-queue wait time from `provision_timeout` accounting on Kubernetes, so clusters waiting in a Kueue queue no longer count against the provisioning timeout budget.

– Named config key but no example values given v0.14.0rc1

03 **Conda removed from default Kubernetes images** BREAKING

65

Conda is no longer installed by default in Kubernetes images; tasks that rely on conda being present in K8s pods will break unless conda is explicitly installed.

– Clear before/after and migration action given v0.14.0rc1

THINNER COVERAGE BELOW

04 **Dashboard Infra page and table UI improvements** IMPROVED

55

The Dashboard Infra page is reworked with a unified table and a per-GPU-type summary, breaks Slurm clusters down by partition, and now shows CPU-only Slurm clusters; Slurm node info is enriched with CPU/memory allocation and sampled usage metrics. Separately, the Dashboard volumes page gains filtering, and the cluster jobs and pools tables now persist the user's rows-per-page preference.

– UI paths named but no exact steps or config v0.14.0rc1

05 **Non-blocking launch request queuing on lock contention**

IMPROVED 55

Launch requests are now parked as `WAITING` on cluster lock contention instead of blocking a worker thread, freeing worker threads for other requests while a cluster lock is held.

– Explains mechanism but no config to control it v0.14.0rc1

06 **Multiple isolated local API servers** NEW 45

SkyPilot now supports running multiple isolated local API servers on one machine, allowing separate SkyPilot API server instances to run without conflicting.

– No flag or config key given for enabling isolation v0.14.0rc1

07 **GCS federated credentials for COPY operations** NEW 45

Adds support for federated credentials for GCS `COPY` operations on GCP, enabling authentication without traditional service-account keys.

– Names GCS COPY op but no config detail v0.14.0rc1

08 **Exit code and log pointer in RECOVERING job events** IMPROVED 45

RECOVERING job events now include the exit code and a log pointer for user-job failure restarts, giving more context on why a job was recovered.

– Names event fields but no API or schema example v0.14.0rc1

09 **On-demand request executor metrics export** NEW 40

Exports metrics for the on-demand request thread executor via the API, exposing executor-level metrics for monitoring.

– No endpoint path or metric names given v0.14.0rc1

10 **Admin-only debug dump creation** BREAKING 35

Restricts creation of debug dumps to admin users only, preventing non-admin accounts from generating support/debug bundles.

– No detail on how admin status is determined v0.14.0rc1

└──▷ **BREAKING ON UPGRADE**

! Conda is no longer installed by default in Kubernetes images — tasks that rely on conda being present in K8s pods will break without explicitly installing it.

v0.14.0rc1 NOTES ↗ PRE-RELEASE

SkyPilot v0.14.0rc1 adds multi-server local API isolation, GCS federated credentials, richer Slurm infra visibility, and Kueue-aware provisioning timeouts.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.14.0rc1 https://github.com/skypilot-org/skypilot.git
# already have the repo? check out this version:
$ git checkout v0.14.0rc1
```

Supports running multiple isolated local API servers on one machine.

Adds SlurmClient.get_all_jobs_info() for per-node job attribution across Slurm clusters.

Supports federated credentials for GCS `COPY` operations on GCP.

Excludes Kueue admission-queue wait time from `provision_timeout` accounting on Kubernetes.

Enriches Slurm node info with CPU/memory allocation and sampled usage metrics.

+9 more

└──▷ BREAKING ON UPGRADE

! Conda is no longer installed by default in Kubernetes images — tasks that rely on conda being present in K8s pods will break without explicitly installing it.

WAS THIS USEFUL? 👍 👎

Eigen Labs Darkbloom

Feedback ⓘ v0.9.8 NOTES ↗ CODE ↗

RANK

Private Inference Network on Idle Macs

> Darkbloom's v0.9.8 release adds native serving for a new multimodal model, recognizes the newest 2026 Mac hardware for reward and capability calculations, extends graceful connection draining to more shutdown and reconnect paths, and hardens retry behavior for Apple's App Attest API.

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

01 **2026 Mac hardware recognition for rewards and serving** NEW

58

Darkbloom recognizes the 2026 M6 and M5 Pro Mac minis and M5 Max Mac Studio for base-reward memory caps and earnings-calculator choices, and the M5 Ultra for serving bandwidth. M6 is reported as its own chip family with conservative MTP and model-capability gates pending physical qualification.

– Names every new chip and its role but no user-facing action v0.9.8

02 **Graceful draining extended to more reconnect paths** IMPROVED

55

Graceful connection draining now covers replacement starts, standalone/foreground handoffs, manual update activation, and planned APNs/model-inventory reconnects, so in-flight requests are not interrupted during these transitions.

– Names the specific transition paths but no config/flag to act on v0.9.8

03 **Bounded retry for Apple App Attest failures** IMPROVED

40

Generic Apple App Attest API failures are now retried using the existing bounded backoff, rather than leaving a live connection permanently pending.

– Describes the fix's effect but not the backoff parameters v0.9.8

04 **Native DiffusionGemma 26B-A4B multimodal serving** NEW

28

Darkbloom adds native serving of the DiffusionGemma 26B-A4B model with multimodal tools.

– Named in summary only, no mechanism or usage detail given v0.9.8

v0.9.8 NOTES ↗ CODE ↗ LATEST STABLE

Darkbloom v0.9.8 adds native DiffusionGemma 26B-A4B serving with multimodal tools, 2026 Mac hardware support, and graceful connection draining.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.9.8 https://github.com/Layr-Labs/d-inference.git
# already have the repo? check out this version:
$ git checkout v0.9.8
```

Extends graceful draining to replacement starts, standalone/foreground handoffs, manual update activation, and planned APNs/model-inventory reconnects so in-flight requests are not interrupted.

Recognizes 2026 M6 and M5 Pro Mac minis and M5 Max Mac Studio for base-reward memory caps and earnings-calculator choices, and M5 Ultra for serving bandwidth.

Reports M6 as its own chip family with conservative MTP and model-capability gates pending physical qualification.

Retries generic Apple App Attest API failures with existing bounded backoff instead of leaving a live connection permanently pending.

WAS THIS USEFUL? 👍 👎

Unsloth AI Unsloth

Feedback ⓘ v0.1.814-beta NOTES ↗

RANK

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

> Across three beta releases, Unsloth added Qwen-Image-2.1 generation, an AMD ROCm Docker image, custom Agent Skills, a native ARM64 desktop package, a 2x faster Thinking UI, and expanded Hugging Face dataset and training-run support in Studio.

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

01 Default GGUF quantization changed to Q4_K_M

BREAKING

75

Changes the default GGUF quantization from F16/BF16 to `Q4_K_M`; existing workflows that relied on the previous default will now download or export a different quantization unless the format is specified explicitly.

— clear before/after with migration note v0.1.814-beta

02 Desktop app packaging for ARM64 and in-app Debian updates

NEW

70

Adds a native ARM64 `.deb` package (`Unsloth-Desktop-Ubuntu-ARM64.deb`) for Ubuntu 24.04+ on ARM64 hardware, and adds in-app updates for Debian installs, removing the need to manually download and reinstall.

Install Unsloth on an Ubuntu 24.04+ ARM64 machine (e.g. an AWS Graviton or Apple M-series Linux box) using the new native ARM64 package.

```
$ curl -fsSL
https://github.com/unslothai/unsloth/releases/latest/download/Uns
loth-Desktop-Ubuntu-ARM64.deb -o Unsloth-Desktop-Ubuntu-ARM64.deb
sudo dpkg -i Unsloth-Desktop-Ubuntu-ARM64.deb
```

— exact package name and install command given v0.1.814-beta v0.1.813-beta

03 Custom Agent Skills in chat NEW

65

Adds custom Agent Skills that can be reused from existing Claude Code and `.agents` folders, selectable via `@` in chat.

— names folders and invocation syntax v0.1.814-beta

04 **Faster reasoning blocks with reworked Thinking UI** IMPROVED

60

Delivers 2x faster long reasoning blocks (60 FPS vs 30 FPS) with a reworked Thinking UI/UX.

– concrete FPS numbers but no user action v0.1.814-beta

THINNER COVERAGE BELOW

05 **Hugging Face datasets in training recipes** NEW

55

Enables Hugging Face datasets in training recipes with correct column mapping for context, question, and answer roles; Hugging Face datasets now work in training recipes.

– names column roles but no config example v0.1.814-beta v0.1.812-beta

06 **Log viewer filtering and navigation improvements** IMPROVED

55

Adds log filtering, long-line wrapping, and a jump-to-latest-entry control in the log viewer.

– names three concrete UI additions v0.1.812-beta

07 **AMD ROCm Docker image with Studio and JupyterLab** NEW

50

Adds a new AMD ROCm Docker image shipping both Unsloth Studio and JupyterLab, matching parity with the existing CUDA image.

– names components but no pull command v0.1.814-beta

08 **Disk cache visibility and clearing in Settings** NEW

50

Adds disk cache visibility and clearing from Settings, so practitioners can see which caches consume space.

– names UI location but no exact path v0.1.814-beta

09 **Draggable chat reordering in sidebar** NEW

50

Chats are now draggable in the sidebar for reordering, pinning, or moving into a project, in both the web UI and Linux desktop app.

– names surfaces but no exact steps v0.1.812-beta

10 **Qwen-Image-2.1 image generation support** NEW

45

Adds Qwen-Image-2.1 support for image editing and image generation.

– names the model but no usage detail v0.1.814-beta

11 **LAN sharing for local model server** NEW 45

Allows other accounts on the network to connect to a local model server when the owner permits, enabling LAN sharing.

— describes behaviour but no config path v0.1.814-beta

12 **Training on gated upstream models via public copies** IMPROVED 45

Gated upstream models can now be trained through their public Unsloth copies without requiring direct access.

— describes capability without example v0.1.812-beta

13 **Vision models see MCP tool images** NEW 40

Enables vision models to see images returned by MCP tools.

— names MCP but no further mechanism v0.1.814-beta

14 **Cleaner Hugging Face Hub model uploads** IMPROVED 40

Model uploads to the Hugging Face Hub now exclude files from earlier exports, keeping pushes clean.

— names Hub uploads but no further detail v0.1.812-beta

15 **Resume training runs from Studio UI** NEW 30

Past training runs can be resumed from the Studio UI.

— bare statement with no mechanism v0.1.812-beta

└──▷ BREAKING ON UPGRADE

! The default GGUF quantization format changed from F16/BF16 to `Q4_K_M`; existing workflows that relied on the previous default will now download or export a different quantization unless the format is specified explicitly.

v0.1.814-beta NOTES ↗ PRE-RELEASE

Unsloth v0.1.814-beta adds Qwen-Image-2.1 support, custom Agent Skills, AMD ROCm Docker image, and 2x faster reasoning blocks.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.814-beta
https://github.com/unslothai/unsloth.git
# already have the repo? check out this version:
$ git checkout v0.1.814-beta
```

└──▷ **TRY IT**

Install Unsloth on an Ubuntu 24.04+ ARM64 machine (e.g. an AWS Graviton or Apple M-series Linux box) using the new native ARM64 package.

```
$ curl -fsSL
https://github.com/unslothai/unsloth/releases/latest/download/Unsloth-Desktop-Ubuntu-ARM64.deb -o Unsloth-Desktop-Ubuntu-ARM64.deb
sudo dpkg -i Unsloth-Desktop-Ubuntu-ARM64.deb
```

Adds native ARM64 Linux `.deb` package (`Unsloth-Desktop-Ubuntu-ARM64.deb`) for Ubuntu 24.04+ on ARM64 hardware.

Adds in-app updates for Debian installs, removing the need to manually download and reinstall.

Adds a new AMD ROCm Docker image shipping both Unsloth Studio and JupyterLab, matching parity with the existing CUDA image.

Adds custom Agent Skills that can be reused from existing Claude Code and `.agents` folders, selectable via `@` in chat.

Adds Qwen-Image-2.1 support for image editing and image generation.

+6 more

└──▷ **BREAKING ON UPGRADE**

! The default GGUF quantization format changed from F16/BF16 to `Q4_K_M`; existing workflows that relied on the previous default will now download or export a different quantization unless the format is specified explicitly.

v0.1.812-beta NOTES ↗ PRE-RELEASE

Unsloth: Hugging Face datasets now work in training recipes, and past training runs can be resumed from the Studio UI.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.1.812-beta
https://github.com/unslothai/unsloth.git
# already have the repo? check out this version:
$ git checkout v0.1.812-beta
```

Adds log filtering, long-line wrapping, and a jump-to-latest-entry control in the log viewer.

> Chats are now draggable in the sidebar for reordering, pinning, or moving into a project (web UI and Linux desktop app).
>
> Gated upstream models can now be trained through their public Unsloth copies without requiring direct access.
>
> Model uploads to the Hugging Face Hub now exclude files from earlier exports, keeping pushes clean.

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore

Feedback ⓘ

changelog · Sep 23, 2026 CHANGELOG ↗ RANK

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

> Amazon Bedrock AgentCore shipped one capability this window: persistent interactive shell sessions inside the AgentCore harness, accessible over a new WebSocket API.

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

01 **Interactive shell sessions in AgentCore harness** NEW

80

Adds the `InvokeAgentRuntimeCommandShell` WebSocket API for persistent, interactive shell sessions inside AgentCore harness microVMs, supporting up to 10 concurrent shells per session and up to 256 KB of buffered output replay on reconnect. These interactive shell sessions work with both the default Harness environment and custom container environments.

— Names API, concurrency limit and buffer size; no code sample given.

changelog-20260923-fe50cca1 Harness: Interactive shells (terminals)…

changelog · Sep 23, 2026 CHANGELOG ↗ LATEST STABLE

AgentCore harness gains interactive WebSocket shells, lifecycle hooks with allow/deny decisions, and custom OpenAI-compatible endpoint routing.

Adds `InvokeAgentRuntimeCommandShell` WebSocket API for persistent, interactive shell sessions inside AgentCore harness microVMs, supporting up to 10 concurrent shells per session and up to 256 KB of buffered output replay on reconnect.

Harness: Interactive shells (terminals) (2026-09-22) BLOG ↗ STABLE

AgentCore Harness gains persistent interactive shell sessions over WebSocket with buffered output replay and reconnect support.

Interactive shell sessions work with both the default Harness environment and custom container environments.

WAS THIS USEFUL? 👍 👎

BerriAI LiteLLM

Feedback ⓘ v1.102.0 NOTES ↗ RANK

LiteLLM is an AI gateway that calls over 100 model APIs in OpenAI format with cost tracking, guardrails, load balancing, and logging.

> LiteLLM v1.102.0 adds infrastructure for multi-path proxy deployments and MCP OAuth relay, alongside new model/provider support, guardrail improvements, and several UI additions for auto-router visibility and budget management.

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

01 Per-request root path resolution via SERVER_ROOT_PATHS NEW

80

A new `SERVER_ROOT_PATHS` environment variable lets the proxy resolve `root_path` per request from a configured prefix list, so a single proxy can be served behind multiple path prefixes (e.g. `/litellm` and `/ai`) and still resolve correctly behind upstream load balancers that route on path prefix.

Serve the proxy behind multiple path prefixes (e.g. `/litellm` and `/ai`) so upstream load balancers routing on path prefix all resolve correctly.

```
$ export SERVER_ROOT_PATHS='/litellm,/ai'
litellm --model gpt-4o
```

— Named env var with a runnable shell example. v1.102.0

02 inspect_embeddings toggle for AIM and Cato guardrails NEW

75

Adds an `inspect_embeddings` toggle for the AIM and Cato guardrail integrations, so embedding requests can also be screened by these guardrails rather than only chat completions.

Enable embedding inspection for the AIM guardrail so that embedding requests are also screened, not just chat completions.

```yaml
guardrails:
  - guardrail_name: aim-guard
    litellm_params:
      guardrail: aim
      inspect_embeddings: true
```

— Named config key with a working YAML example. v1.102.0

03 **lite pi CLI subcommand** NEW 65

Adds a `lite pi` CLI subcommand to run the Pi coding agent through the LiteLLM proxy.

Run the Pi coding agent through the LiteLLM proxy without any extra setup.

```
$ lite pi
```

— Named command runnable as shown, but no further mechanism given. v1.102.0

04 **New model and API support: Mistral TTS, Vertex Lyria, Fireworks Responses API**

NEW 60

Adds text-to-speech support for Mistral via `/v1/audio/speech`, adds Vertex AI Lyria model support, and adds native Responses API config for Fireworks AI.

— Names the endpoint and providers but no usage example. v1.102.0

THINNER COVERAGE BELOW

05 **Auto-router usage and setup visibility in the UI** NEW 50

Adds a key-scoped auto-router usage tab in the UI showing classification spend itemized by auto-router decisions, plus a UI panel that makes automatic auto-router setup discoverable and displays what it configured.

— Describes two UI additions but only a navigation starting point. v1.102.0

06 **Post-call guardrails on streaming responses** IMPROVED 45

The policy engine now executes post-call guardrail pipelines on streaming responses.

— Explains the behavior change but gives no config example. v1.102.0

07 **Cost map pricing updates for Bedrock GovCloud and Azure AI Foundry**

IMPROVED 40

Adds GovCloud pricing for live but previously unpriced Bedrock models, and adds `azure_ai/gpt-6-astra` Foundry pricing to the cost map.

— Names a specific model id but no way to act on it directly. v1.102.0

08 **Edit soft budget on virtual keys** IMPROVED 40

Adds the ability to edit soft budget on existing virtual keys.

— Names the affected object but not the UI or API path. v1.102.0

09 **ChatGPT subscription provider option in Add Model form** NEW 40

Adds a ChatGPT subscription provider option in the Add Model form in the UI.

— Names the UI form as a clear starting point. v1.102.0

10 **MCP OAuth relay discovery** NEW 30

Adds opt-in per-server OAuth relay discovery for MCP servers.

— Named capability but no setup detail or command given. v1.102.0

11 **Semantic search over skill registry** NEW 25

Adds semantic search over the LiteLLM-hosted skill registry.

— Bare description with no mechanism or access path. v1.102.0

12 **Heuristic v1 tuning gate for router** NEW 25

Adds a heuristic v1 tuning gate for the router.

— No explanation of what the gate tunes or how to use it. v1.102.0

v1.102.0 NOTES ↗ LATEST STABLE

LiteLLM v1.102.0 adds MCP OAuth relay, Mistral TTS, Vertex Lyria, Fireworks Responses API, and per-request root path resolution via `SERVER_ROOT_PATHS`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.102.0 https://github.com/BerriAI/litellm.git
# already have the repo? check out this version:
$ git checkout v1.102.0
```

└──▷ TRY IT

Run the Pi coding agent through the LiteLLM proxy without any extra setup.

```
$ lite pi
```

Serve the proxy behind multiple path prefixes (e.g. `/litellm` and `/ai`) so upstream load balancers routing on path prefix all resolve correctly.

```
$ export SERVER_ROOT_PATHS='/litellm,/ai'
litellm --model gpt-4o
```

Enable embedding inspection for the AIM guardrail so that embedding requests are also screened, not just chat completions.

```yaml
guardrails:
  - guardrail_name: aim-guard
    litellm_params:
      guardrail: aim
      inspect_embeddings: true
```

Adds `SERVER_ROOT_PATHS` environment variable to resolve `root_path` per request from a configured prefix list in the proxy.

Adds `inspect_embeddings` toggle for AIM and Cato guardrail integrations.

Adds opt-in per-server OAuth relay discovery for MCP servers.

Adds `lite pi` CLI subcommand to run the Pi coding agent through the proxy.

Adds text-to-speech support for Mistral via `/v1/audio/speech`.

+11 more

WAS THIS USEFUL?  

Microsoft ONNX Runtime

Feedback ⓘ

plugin-ep-webgpu/v0.4.0 · NOTES ↗ · RANK

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

> ONNX Runtime's WebGPU Plugin EP v0.4.0 adds Linux AArch64 packages, deepens PagedAttention inference with int8 KV cache quantization and GPT-OSS support, introduces new DeepSeek Engram contrib ops, and rolls out a broad set of WebGPU kernel and pipeline performance optimizations.

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

01 WebGPU kernel and pipeline performance optimizations IMPROVED

75

A set of WebGPU execution optimizations: vectorizes `Split` when every output segment is vec4-aligned; optimizes `MatMulNBits` wide-tile execution using subgroup shuffle; fuses eight additional activations into Conv and supplies fused activation parameters as uniforms for Conv/MatMul; adds fused activation support to the im2col Conv path; scales Dawn pipeline compilation workers with CPU count; prepacks Conv weights for the im2col-matmul path to reduce per-inference overhead; selects the pooling path by occupancy rather than output size; and shares subgroup matrix MatMul with pointwise Conv while pinning subgroup size to 32 for subgroup-matrix MatMul/Gemm.

– Names every optimized op and mechanism, no benchmark numbers plugin-ep-webgpu/v0.4.0

02 DeepSeek Engram contrib ops on WebGPU NEW

65

Adds `EngramGate` and `NGramHashMapping` contrib op kernels for DeepSeek Engram support on WebGPU.

– Names exact new ops but no usage example plugin-ep-webgpu/v0.4.0

03 PagedAttention inference enhancements IMPROVED

60

Adds int8 KV cache block quantization for PagedAttention inference, plus PagedAttention metadata and GPT-OSS support, with continued PagedAttention optimization work.

– Names int8 quantization and GPT-OSS but no config/flag given plugin-ep-webgpu/v0.4.0

THINNER COVERAGE BELOW

04 WebGPU subgroup-size-control infrastructure NEW

50

Implements WebGPU subgroup-size-control infrastructure and enables the subgroup matrix path for WASM builds.

— States what was added, mechanism thin `plugin-ep-webgpu/v0.4.0`

05 **Linux AArch64 packages for WebGPU Plugin EP** NEW 35

Adds Linux AArch64 packages for the WebGPU Plugin EP.

— Only stated in summary, no install detail given `plugin-ep-webgpu/v0.4.0`

`plugin-ep-webgpu/v0.4.0` NOTES ↗ LATEST STABLE

ONNX Runtime WebGPU Plugin EP v0.4.0 adds Linux AArch64 packages, int8 KV cache quantization, and new contrib ops including EngramGate and NGramHashMapping.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch plugin-ep-webgpu/v0.4.0
https://github.com/microsoft/onnxruntime.git
# already have the repo? check out this version:
$ git checkout plugin-ep-webgpu/v0.4.0
```

Adds `EngramGate` and `NGramHashMapping` contrib op kernels for DeepSeek Engram support on WebGPU.

Adds int8 KV cache block quantization for PagedAttention inference.

Adds PagedAttention metadata and GPT-OSS support with continued PagedAttention optimization.

Implements WebGPU subgroup-size-control infrastructure and enables the subgroup matrix path for WASM builds.

Vectorizes `Split` when every output segment is vec4-aligned for improved throughput.

+7 more

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.64 NOTES ↗ RANK

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

> HyperFrames shipped a new seek-deterministic `data-vfx-chain` GPU shader effect element, added IBM Bob as a supported CLI coding agent, exposed new Studio component exports, and polished Studio's timeline visuals with transition badges, real audio peak meters, and accurate filmstrip geometry.

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

01 **data-vfx-chain shader effect element** NEW 90

Adds `data-vfx-chain`, a seek-deterministic per-pixel effect element that runs GPU shader chains — including fractal noise, wave warp, and displacement map — computed from seek time alone so renders and previews match exactly.

– Names the element and its shader effects but gives no usage example. v0.8.64

02 **New Studio component exports** NEW 60

Studio now exports `AudioWaveform`, `ImageThumbnail`, and `useRenderClipContent` for use in custom Studio-based integrations.

– Names three concrete exports but no usage example or behavior. v0.8.63

THINNER COVERAGE BELOW

03 **Studio timeline visual improvements** IMPROVED 55

Studio transition seams now display a soft blur and carry a badge marking cut points in the timeline; audio pills show real peak and loudness waveforms instead of placeholder visuals; and video filmstrips render at their true geometry size for accurate timeline scrubbing.

– Describes three UI changes without exact navigation or screenshot. v0.8.61

04 **IBM Bob coding agent support** NEW 40

Adds IBM Bob runtime support to the CLI as a supported coding agent.

– Names the agent but no flag, command, or setup detail. v0.8.63

v0.8.64 NOTES ↗ LATEST STABLE

HyperFrames v0.8.64 adds `data-vfx-chain`, a seek-deterministic per-pixel GPU shader chain element.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.8.64 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.64
```

v0.8.63 NOTES ↗ STABLE

HyperFrames v0.8.63 adds IBM Bob coding agent support and exports AudioWaveform, ImageThumbnail, and useRenderClipContent from Studio.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.63 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.63
```

v0.8.61 NOTES ↗ STABLE

Studio gains visual transition badges, real audio peak meters, and geometry-accurate video filmstrips in v0.8.61.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.61 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.61
```

Studio transition seams now display a soft blur and carry a badge, making cut points visually identifiable in the timeline.

Audio pills in Studio display real peak and loudness waveforms instead of placeholder visuals.

Video filmstrips in Studio render at their true geometry size for accurate timeline scrubbing.

WAS THIS USEFUL? 👍 👎

Ollama

Feedback ⓘ v0.34.3 NOTES ↗ CODE ↗ RANK

Local large language model runner that downloads, serves, and chats with models through a CLI and REST API.

> Ollama v0.34.3 lets clients discover a model's supported reasoning modes via API and CLI, and adds MLX support for Nemotron H vision models on Apple Silicon.

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

01 **Thinking-level discovery via API and CLI** NEW 86

`GET /api/show` now returns a `thinking` object with `values` (e.g. `[false, true]` or `['low', 'high', 'max']`) and `default`, letting clients discover each model's supported reasoning controls at runtime. The `ollama show <model>` CLI command displays the same `levels` and `default` under its Capabilities section for models that support it.

Inspect a cloud model's supported thinking levels before choosing a reasoning mode for your request.

```
$ curl http://localhost:11434/api/show -d '{"model": "glm-5.3-flash:cloud"}'
```

Check thinking levels and defaults for a local model directly from the CLI.

```
$ ollama show gemma4
```

— Names exact endpoint, field names, and CLI command v0.34.3

THINNER COVERAGE BELOW

02 **Nemotron H vision models via MLX on Apple Silicon** NEW 50

Adds support for Nemotron H vision models on Apple Silicon via MLX, including the RADIO vision encoder and dynamic-resolution preprocessing.

— Names model and mechanism but no usage steps v0.34.3

v0.34.3 NOTES ↗ CODE ↗ LATEST STABLE

> `GET /api/show` now exposes per-model thinking levels and defaults; Nemotron H vision models gain MLX support on Apple Silicon.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.34.3 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.34.3
```

└──▷ **TRY IT**

Inspect a cloud model's supported thinking levels before choosing a reasoning mode for your request.

```
$ curl http://localhost:11434/api/show -d '{"model": "glm-5.3-flash:cloud"}'
```

Check thinking levels and defaults for a local model directly from the CLI.

```
$ ollama show gemma4
```

Extends `GET /api/show` to return a `thinking` object with `values` (e.g. `[false, true]` or `['low', 'high', 'max']`) and `default`, letting clients discover each model's supported reasoning controls at runtime.

Extends `ollama show <model>` CLI output to display thinking `levels` and `default` under the Capabilities section for models that support it.

Supports Nemotron H vision models on Apple Silicon via MLX, including the RADIO vision encoder and dynamic-resolution preprocessing.

WAS THIS USEFUL? 👍 👎

ZenML

Feedback ⓘ 0.97.0 NOTES ↗ RANK

ZenML is an MLOps and LLMOps framework for building portable pipelines that run the same code across local, cloud, and orchestrator backends.

> ZenML 0.97.0 introduces resource pools v2 for more granular compute allocation, alongside improvements to log store querying and API transaction handling.

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

01 **Resource pools v2 for compute allocation** NEW 30

Adds support for resource pools v2, enabling more granular compute resource allocation for pipeline steps.

— Describes benefit but no config keys or mechanism 0.97.0

02 **Pagination and filtering for external log stores** IMPROVED 28

Adds pagination and filtering for external log stores, making it practical to query large log datasets.

— General description, no API or flag names given 0.97.0

03 **Larger API transaction support** IMPROVED 18

Allows larger API transactions, raising the ceiling on payload sizes for API calls.

— No concrete limit or endpoint specified 0.97.0

0.97.0 NOTES ↗ LATEST STABLE

ZenML 0.97.0 adds resource pools v2, pagination/filtering for external log stores, and larger API transaction support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 0.97.0 https://github.com/zenml-io/zenml.git
# already have the repo? check out this version:
$ git checkout 0.97.0
```

Adds support for resource pools v2, enabling more granular compute resource allocation for pipeline steps.

Adds pagination and filtering for external log stores, making it practical to query

> large log datasets.
>
> Allows larger API transactions, raising the ceiling on payload sizes for API calls.

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.23.0 NOTES ↗

RANK

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

> OpenAI TypeScript SDK v7.23.0 adds support for GCP external storage and introduces the GPT-Rosalind research model to the client's supported model list.

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

01 **GPT-Rosalind research model added** NEW 35

Adds `gpt-rosalind` research model to the supported model list.

— Named model added, no usage detail given v7.23.0

02 **GCP external storage support** NEW 25

The SDK adds support for GCP external storage, according to the release summary, though no further mechanism or configuration details are given.

— Named but undetailed in source v7.23.0

v7.23.0 NOTES ↗ LATEST STABLE

OpenAI TypeScript SDK v7.23.0 adds GCP external storage support and the GPT-Rosalind research model.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v7.23.0 https://github.com/openai/openai-node.git
# already have the repo? check out this version:
$ git checkout v7.23.0
```

WAS THIS USEFUL? 👍 👎

LLM Compressor

Feedback ⓘ 0.14.0 NOTES ↗ RANK

LLM Compressor quantizes and compresses language models for deployment with vLLM and other inference engines.

> LLM Compressor's 0.14.0 window centers on REAP MoE expert pruning, a new unified ModelFreePtqConverter for model-free PTQ, and a Triton-based GPTQ kernel roughly 15x faster, alongside NVFP4 quantization support for the GLM-5.3-Flash vision-language MoE model and support for several new model families.

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

01 **REAP MoE expert pruning** NEW 95

`REAPPruningModifier` (from `llmcompressor.modifiers.pruning`) structurally removes low-saliency experts from Mixture-of-Experts layers, with `sparsity` controlling the fraction of experts removed per layer, `report_path` (e.g. `'reap_retained_experts.json'`) recording which expert indices are retained, and `moe_calibrate_all_experts=False` enabling a faster pruning-only run that uses only routed experts for saliency. It can be combined with quantization modifiers (including FP8/NVFP4 recipes) in the same recipe list, now supports distributed DDP runs where saliency statistics are reduced across ranks and broadcast back to workers, and gained an optional e-score correction bias.

Prune 50% of MoE experts by saliency and save which experts were retained, without calibrating all experts (faster pruning-only run).

python

```python
from llmcompressor.modifiers.pruning import REAPPruningModifier

recipe = REAPPruningModifier(
    sparsity=0.5,
    report_path='reap_retained_experts.json',
    moe_calibrate_all_experts=False,
)
```

Reduce VRAM usage of a MoE model by pruning the least-salient experts per layer, writing a saliency report, then validating before deploying.

```python
from llmcompressor import oneshot
from llmcompressor.modifiers.pruning import REAPPruningModifier

oneshot(
    model=model,
    dataset="perfectblend",
    recipe=REAPPruningModifier(
        sparsity=0.2,
        report_path="reap.json",
        moe_calibrate_all_experts=False,
    ),
    num_calibration_samples=512,
    max_seq_length=2048,
)
```

— Names flags and params with two runnable recipe examples `0.14.0`

02 **NVFP4 quantization for GLM-5.3-Flash** NEW 95

NVFP4 quantization now supports the GLM-5.3-Flash vision-language MoE model: `QuantizationModifier` with `scheme='NVFP4'` targets the MoE expert `gate`, `up`, and `down` projections while `ignore` keeps the vision tower, `lm_head`, `mlp.gate` router, and attention indexer at full precision. `FP8BlockDequantizer` via `convert_checkpoint` first decompresses GLM-5.3-Flash's native FP8-block checkpoint to bfloat16; calibration can be distributed across GPUs with `torchrun --nproc-per-node N` using `device_map='auto_offload'` for CPU offloading, and a new `load_context` utility loads `Glm5NextForConditionalGeneration` to preserve the vision tower instead of the generic `AutoModelForCausalLM`. General quantization support for GLM 5.3 and GLM 5.3 Flash was also added.

Dequantize the native FP8-block GLM-5.3-Flash checkpoint to bfloat16 before applying NVFP4 quantization.

```python
from compressed_tensors.entrypoints.convert import
FP8BlockDequantizer, convert_checkpoint

MODEL_ID = "zai-org/GLM-5.3-Flash"
SAVE_DIR = MODEL_ID.rstrip("/").split("/")[-1] + "-BF16"

convert_checkpoint(
    model_stub=MODEL_ID,
    save_directory=SAVE_DIR,
    converter=FP8BlockDequantizer.from_pretrained(MODEL_ID),
    max_workers=8,
)
```

Quantize GLM-5.3-Flash MoE expert weights to NVFP4 with distributed calibration across multiple GPUs, keeping the vision tower and router at full precision.

```
$ torchrun --nproc-per-node 8 glm53_flash_example.py
```

Configure the NVFP4 recipe to target only MoE expert projections while excluding vision tower, lm_head, router, and attention indexer layers.

```python
from llmcompressor.modifiers.quantization import
QuantizationModifier

recipe = QuantizationModifier(
    scheme="NVFP4",
    targets=[r"re:.*mlp\.experts\..*(gate|up|down)_proj$"],
    ignore=[
        r"re:.*visual.*",
        "lm_head",
        r"re:.*mlp\.gate$",
        r"re:.*self_attn\.indexer\..*",
    ],
)
```

— Full model-specific recipe with commands and exact targets `0.14.0`

03 **MoE checkpoint mapping helpers** NEW 85

patch_moe_mappings() overrides a checkpoint's 2D expert load mappings on a per-checkpoint basis through a load context, for custom or non-standard MoE layouts, while

repack_moe() restores native fused 3D expert modules after linearization so save_pretrained() writes Hugging Face-native keys.

Restore native fused 3D expert modules in an MoE checkpoint after linearization, then override its 2D load mappings for a custom checkpoint layout.

```python
from llmcompressor.utils.moe import patch_moe_mappings,
repack_moe

# Restore fused expert modules after linearization
repack_moe(model)

# Override 2D load mappings for a non-standard checkpoint
patch_moe_mappings(model, mappings={'experts.0.w1':
'experts.0.gate_proj'})
```

After linearizing a MoE checkpoint, use repack_moe() to restore native fused 3D expert modules before saving so vLLM can load the result without remapping.

```python
from llmcompressor.transformers.compression.helpers import
repack_moe

repack_moe(model)
model.save_pretrained('my-moe-model-repacked')
tokenizer.save_pretrained('my-moe-model-repacked')
```

— Two named functions with runnable usage shown 0.14.0

04 **Triton kernels for GPTQ and MSE observer** IMPROVED 85

A new Triton-based GPTQ quantization kernel is approximately 15x faster end-to-end than the previous eager path (measured on Llama-3-8B), with layer batching delivering a further ~2x speedup on MoE models (~30x total on Qwen3-30B-A3B). A Triton kernel was also added to the MSE observer's scale grid search, using buffered per-qparam patience and adaptive 512-value tiling for roughly 10x faster search, supporting INT, FP4, FP8, and FP16/BF16 with E8M0 scales; the new `triton_error_buffer` parameter controls approximate congruence between eager and Triton patience behavior, defaulting to 100% for FP4 and 30% otherwise.

— Concrete speedups and a named config parameter, no runnable snippet 0.14.0

05 **Breaking changes in 0.14.0** BREAKING 80

0.14.0 removes the deprecated GPTQ Group/Dynamic Activation Ordering and removes `GraniteMoeLinearExperts`. Model-free PTQ is now unified under `ModelFreePtqConverter`, consolidating previous entry points. `TensorProfiler` must now be imported from `compressed-tensors` instead of `llmcompressor`, and `exec_jobs_dynamic` has moved to `compressed-tensors` and is no longer available from `llmcompressor`. The default for `pad_to_max_length` has changed, and the minimum version of `accelerate` is now `>=1.15.0`.

— Enumerates every removed/changed surface, a clear migration list `0.14.0`

06 **ModelFreePtqConverter for model-free PTQ** NEW 70

`ModelFreePtqConverter` is a new unified entry point for model-free PTQ, consolidating mixed-precision and KV-cache quantization support. It replaces the previous static round-robin GPU assignment with a dynamic, memory-aware scheduler that queries available GPU memory before each job, and adds profiler-based memory estimates for model-free PTQ runs.

— Names the class and mechanism but no example call `0.14.0`

07 **New model family support** NEW 70

Adds fast-loading support and 2D conversion mappings for `nemotron_h` (Nemotron 3 Ultra), GPT-OSS expert linearization support in the MoE machinery, a native `KimiK3ForConditionalGeneration` model definition with upfront model decompression and quantization examples, and AWQ/SmoothQuant mappings for `OlmoForCausalLM` v1/v2 using the Exaone4-style mapping.

— Names four model families but no code shown `0.14.0`

08 **AutoRound modifier and mixed-precision recipe examples** NEW 70

`AutoRound` is now a supported modifier alongside GPTQ, AWQ, SmoothQuant, and REAP in the `oneshot` entrypoint, and its calibration inputs can be offloaded to CPU for large `N` to reduce VRAM usage. New examples ship combining QuIP + GPTQ + NVFP4A16 (MR-GPTQ) and FP8 Attention + AutoRound for Qwen3 dense and MoE models, alongside a mixed W2A16/W4A16 MoE example, a GLM-5.2 MXFP4\u00d7MXFP8 example, and a DeepSeek V4 MXFP4\u2013MXFP8 example.

— Names modifier and five example recipes without runnable snippets `0.14.0`

09 **NVFP4 accuracy observers** NEW 60

New `nvfp4_expanded_mse` and `nvfp4_expanded_imatrix` observers improve accuracy for NVFP4 and NVFP4A16 quantization, using insights from the Four Over Six approach. The MSE and iMatrix observers gained a grid-search expansion factor making the search space a strict superset of the Fourosix strategy, enabling better local scales for NVFP4.

— Names the observers but gives no usage path 0.14.0

10 **Prebaked calibration datasets** NEW 60

Adds the `perfectblend` prebaked calibration dataset and an `ultrachat` alias for use with oneshot().

Use the `perfectblend` prebaked dataset for calibration when running oneshot(), replacing a manually constructed dataset.

python

```
from llmcompressor import oneshot
from llmcompressor.modifiers.quantization import
QuantizationModifier

recipe = QuantizationModifier(targets='Linear',
scheme='FP8_DYNAMIC', ignore=['lm_head'])

oneshot(
    model=model,
    dataset='perfectblend',
    recipe=recipe,
    max_seq_length=2048,
    num_calibration_samples=512,
)
```

— Named dataset with a runnable oneshot() example 0.14.0

THINNER COVERAGE BELOW

11 **RTN fallback summary for GPTQ** NEW 40

Adds an RTN fallback end-of-run summary listing modules that fell back to RTN during GPTQ quantization.

— Names the feature but no output format or example 0.14.0

12 **Config preservation on save** NEW 35

Adds `resave_config` to preserve the original model configuration on save.

— Bare one-line description, no usage shown 0.14.0

└──▷ **BREAKING ON UPGRADE**

! Deprecated GPTQ Group/Dynamic Activation Ordering has been removed.

! `GraniteMoeLinearExperts` has been removed.

! Model-free PTQ is now unified under `ModelFreePtqConverter`; previous entry points are consolidated.

! `TensorProfiler` must now be imported from `compressed-tensors` instead of `llmcompressor`.

! `exec_jobs_dynamic` has been moved to `compressed-tensors` and is no longer available from `llmcompressor`.

! The default for `pad_to_max_length` has changed.

! The minimum version of `accelerate` is now `>=1.15.0`.

0.14.0 NOTES ↗ LATEST STABLE

LLM Compressor 0.14.0: Triton GPTQ kernel (~15× faster), new ModelFreePtqConverter, MoE helpers, and 9 new model families

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch 0.14.0 https://github.com/vllm-project/llm-compressor.git
# already have the repo? check out this version:
$ git checkout 0.14.0
```

└──▷ **USE IT**

After linearizing a MoE checkpoint, use repack_moe() to restore native fused 3D expert modules before saving so vLLM can load the result without remapping.

```python
from llmcompressor.transformers.compression.helpers import repack_moe

repack_moe(model)
model.save_pretrained('my-moe-model-repacked')
tokenizer.save_pretrained('my-moe-model-repacked')
```

Use the `perfectblend` prebaked dataset for calibration when running oneshot(), replacing a manually constructed dataset.

```python
from llmcompressor import oneshot
from llmcompressor.modifiers.quantization import
QuantizationModifier

recipe = QuantizationModifier(targets='Linear',
scheme='FP8_DYNAMIC', ignore=['lm_head'])

oneshot(
    model=model,
    dataset='perfectblend',
    recipe=recipe,
    max_seq_length=2048,
    num_calibration_samples=512,
)
```

Adds `ModelFreePtqConverter` as the unified entry point for model-free PTQ, consolidating mixed-precision and KV-cache quantization support.

Adds patch_moe_mappings() to override 2D load mappings on a per-checkpoint basis through a load context.

Adds repack_moe() to restore native fused 3D expert modules so save_pretrained() writes Hugging Face-native keys.

Adds `triton_error_buffer` to the MSE observer to control approximate congruence between eager and Triton patience behavior, defaulting to 100% for FP4 and 30% otherwise.

Adds `resave_config` to preserve the original model configuration on save.

+17 more

└──▷ **BREAKING ON UPGRADE**

- ! Deprecated GPTQ Group/Dynamic Activation Ordering has been removed.
- ! `GraniteMoeLinearExperts` has been removed.
- ! Model-free PTQ is now unified under `ModelFreePtqConverter`; previous entry points are consolidated.
- ! `TensorProfiler` must now be imported from `compressed-tensors` instead of `llmcompressor`.
- ! `exec_jobs_dynamic` has been moved to `compressed-tensors` and is no longer available from `llmcompressor`.
- ! The default for `pad_to_max_length` has changed.
- ! The minimum version of `accelerate` is now `>=1.15.0`.

WAS THIS USEFUL? 👍 👎

Anthropic TypeScript SDK Feedback ⓘ sdk-v0.128.0

NOTES ↗ RANK

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

> The Anthropic TypeScript SDK added support for the claude-opus-5-5 model in this release.

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

01 **claude-opus-5-5 model support** NEW 55

The SDK adds support for the `claude-opus-5-5` model, usable via the `model` field in `client.messages.create` and agent/session APIs.

— Names exact field and model but no other mechanism detail. sdk-v0.128.0

sdk-v0.128.0 NOTES ↗ LATEST STABLE

Adds claude-opus-5-5 model support, inline tool definitions, and MCP tool-list pinning (beta) to the TypeScript SDK.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch sdk-v0.128.0
https://github.com/anthropics/anthropic-sdk-typescript.git
# already have the repo? check out this version:
$ git checkout sdk-v0.128.0
```

Adds support for the `claude-opus-5-5` model, usable via the `model` field in `client.messages.create` and agent/session APIs.

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ v1.8.0 NOTES ↗ RANK

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

> The Anthropic Python SDK's v1.8.0 release adds support for the claude-opus-5-5 model, lets tools be defined inline in API calls, and introduces a beta MCP tool-list pinning feature.

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

01 **claude-opus-5-5 model support** NEW 75

Adds `claude-opus-5-5` as a usable model, selectable via client.messages.create(model=...), as demonstrated by a call to client.messages.create(model="claude-opus-5-5", max_tokens=1024, messages=[...]).

Use the new claude-opus-5-5 model for a higher-capability inference call.

python

```python
import os
from anthropic import Anthropic

client = Anthropic(api_key=os.environ.get("ANTHROPIC_API_KEY"))

message = client.messages.create(
    model="claude-opus-5-5",
    max_tokens=1024,
    messages=[{"role": "user", "content": "Summarize the OWASP
Top 10 for 2025."}],
)
print(message.content)
```

— Runnable example shown, but no further mechanism detail. v1.8.0

THINNER COVERAGE BELOW

02 **MCP tool-list pinning (beta)** NEW 45

Adds a beta MCP tool-list pinning capability, enabling callers to lock down which MCP tools are available within a session.

— Described only in prose, no API or flag named. v1.8.0

03 **Inline tool definitions in API calls** NEW 40

Tools can now be specified directly within an API call rather than requiring pre-registration, allowing callers to define tools inline per request.

— No code sample or parameter name given. v1.8.0

v1.8.0 NOTES ↗ LATEST STABLE

Adds claude-opus-5-5 model support, inline tool definitions, and MCP tool-list pinning (beta) to the Python SDK.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.8.0 https://github.com/anthropics/anthropic-sdk-python.git
# already have the repo? check out this version:
$ git checkout v1.8.0
```

└──▷ USE IT

Use the new claude-opus-5-5 model for a higher-capability inference call.

python

```
import os
from anthropic import Anthropic

client = Anthropic(api_key=os.environ.get("ANTHROPIC_API_KEY"))

message = client.messages.create(
    model="claude-opus-5-5",
    max_tokens=1024,
    messages=[{"role": "user", "content": "Summarize the OWASP Top 10 for 2025."}],
)
print(message.content)
```

Adds support for `claude-opus-5-5` as a usable model in client.messages.create(model=...).

Adds inline tool definitions, allowing tools to be specified directly in API calls rather than pre-registered.

Adds MCP tool-list pinning (beta), enabling callers to lock down which MCP tools are available in a session.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

Mintplex Labs AnythingLLM Feedback ⓘ v1.16.2

NOTES ↗ RANK

AnythingLLM is a local-first desktop and server application that runs agents, RAG workspaces, and MCP tools against local or hosted models.

> AnythingLLM v1.16.2 broadens its provider ecosystem with new LLM and web-search providers, adds a configurable agent tool-call approval timeout, and ships smaller improvements to agent web browsing, image generation, embeddings, and web scraping.

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

01 Configurable agent tool-call approval timeout NEW

80

Adds the `TOOL_CALL_APPROVAL_TIMEOUT_MS` environment variable to the admin env config for controlling how long AnythingLLM waits for user approval on agent tool calls before timing out.

Set a custom timeout for agent tool-call approvals so long-running automations don't stall indefinitely waiting for user confirmation.

env

```
TOOL_CALL_APPROVAL_TIMEOUT_MS=120000
```

— Exact env var named with a runnable example value v1.16.2

02 Google Vertex AI and LLMman as LLM providers NEW

60

Adds Google Vertex AI as an LLM provider, and adds LLMman (OSS) as a local LLM provider, replacing Docker Model Runner.

— Names both providers and the replacement, no setup steps v1.16.2

03 New default and additional web search providers NEW

60

Adds You.com as the default web search provider, with Zero Data Retention, and adds Keenable and AnySearch as additional web search provider options.

— Names all three providers and the ZDR detail, no config steps v1.16.2

THINNER COVERAGE BELOW

04 GenieX model management on Windows on ARM IMPROVED

55

GenieX on Windows on ARM now lists models from your local model directory and supports pulling models directly from Hugging Face in the UI.

— Names platform and capability, no exact UI path v1.16.2

05 **generate-image agent skill** NEW 50

Adds a `generate-image` agent skill for creating images directly from chat.

— Names the skill but gives no usage mechanism v1.16.2

06 **OpenRouter service tier selection** NEW 45

Adds OpenRouter service tier selection for controlling routing cost and priority.

— Names the control but not the tier options or setting v1.16.2

07 **Quick access to logs via version tag** IMPROVED 45

Clicking the version tag at the top of the window now jumps to logs and storage directory.

— Exact UI navigation described v1.16.2

08 **Batched Mistral embeddings** IMPROVED 40

Mistral embeddings are now batched, making large document embedding significantly faster.

— States mechanism and benefit but no numbers v1.16.2

09 **Structured web scraping output** IMPROVED 40

Scraped web pages now preserve tables, ordered lists, and code blocks instead of flattening them.

— Clear before/after but no config or limits v1.16.2

10 **LocalAI image editing via reference images** NEW 30

Adds LocalAI image editing via reference images.

— Bare one-line description, no mechanism given v1.16.2

11 **Hindi, Bulgarian, and Swedish language support** NEW 30

Adds Hindi, Bulgarian, and Swedish language support.

— Names languages but not scope of coverage v1.16.2

12 **Optional arguments in agent flow variables** IMPROVED 25

Adds agent flow variable support for optional arguments.

— Single thin line with no example of use v1.16.2

13 **Web browsing enabled by default for agents** IMPROVED

25

Web browsing is now enabled by default for agents.

— States the change with no configuration or toggle detail v1.16.2

v1.16.2 NOTES ↗ LATEST STABLE

AnythingLLM v1.16.2 adds Google Vertex AI, LLMman, a `generate-image` agent skill, You.com default search, and a `TOOL_CALL_APPROVAL_TIMEOUT_MS` env var.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.16.2 https://github.com/Mintplex-Labs/anything-llm.git
# already have the repo? check out this version:
$ git checkout v1.16.2
```

└──▷ **USE IT**

Set a custom timeout for agent tool-call approvals so long-running automations don't stall indefinitely waiting for user confirmation.

.env

```
TOOL_CALL_APPROVAL_TIMEOUT_MS=120000
```

Adds `TOOL_CALL_APPROVAL_TIMEOUT_MS` environment variable to admin env config for controlling agent tool-call approval timeouts.

Adds `generate-image` agent skill for creating images directly from chat.

Adds Google Vertex AI as an LLM provider.

Adds LLMman (OSS) as a local LLM provider, replacing Docker Model Runner.

Adds You.com as the default web search provider, with Zero Data Retention.

+11 more

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Neo4j Agent Memory

Feedback ⓘ nams-ai-provider-v0.3.0

CODE ↗ RANK

Agent Memory is a graph-native memory system for AI agents that stores conversations, builds knowledge graphs, and tracks reasoning traces with Neo4j.

> Agent Memory's TypeScript provider now injects Neo4j graph relationships directly into prompts and adds a full hooks-based lifecycle API for controlling session memory.

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

01 **Hooks lifecycle API via createNams().hooks()** NEW 90

New createNams().hooks() mode exposes loadSession(), prepare(), withHooks(), onFinish(), and end() methods, letting the runtime rather than the model control when session memory loads and persists. It supports lifecycle hooks for `SessionStart`, `UserPromptSubmit`, `PreToolUse`, `PostToolUse`, `PostToolUseFailure`, `PreMemoryWrite`, `Stop`, and `SessionEnd`.

Use hooks mode so your runtime — not the model — controls when session memory is loaded and persisted across every generation.

typescript

```typescript
const session = createNams().hooks();
const messages = await session.loadSession(scope);
// ... call generateText with prepended messages ...
await session.onFinish(scope);
```

— Full method/hook list plus a runnable code example nams-ai-provider-v0.3.0

02 **Graph retrieval expansion in AI provider** NEW 87

After entity search, matched entities are expanded and their Neo4j relationships are injected into the prompt as triples (e.g. (Alex)-[WORKS_AT]->(TechCorp)) tagged with `source: 'graph'`. Expansion is controlled by `graphExpansionLimit` (default: 2 entities, `0` disables it), and is capped at five relationships per entity.

— Names config key, defaults and cap; no runnable snippet given nams-ai-provider-v0.3.0

nams-ai-provider-v0.3.0 CODE ↗ LATEST STABLE

Agent Memory TypeScript provider gains graph retrieval with `graphExpansionLimit` and a new createNams().hooks() lifecycle API

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch nams-ai-provider-v0.3.0
https://github.com/neo4j-labs/agent-memory.git
# already have the repo? check out this version:
$ git checkout nams-ai-provider-v0.3.0
```

└──▷ **USE IT**

Use hooks mode so your runtime — not the model — controls when session memory is loaded and persisted across every generation.

```typescript
const session = createNams().hooks();
const messages = await session.loadSession(scope);
// ... call generateText with prepended messages ...
await session.onFinish(scope);
```

Adds graph retrieval to the Vercel AI provider: after entity search, matched entities are expanded and their Neo4j relationships are injected into the prompt as triples (e.g. (Alex)-[`WORKS_AT`]->(TechCorp)) with `source: 'graph'`; controlled by `graphExpansionLimit` (default: 2 entities, `0` disables it), capped at five relationships per entity.

Adds createNams().hooks() hooks mode with loadSession(), prepare(), withHooks(), onFinish(), and end() methods, plus lifecycle hooks for `SessionStart`, `UserPromptSubmit`, `PreToolUse`, `PostToolUse`, `PostToolUseFailure`, `PreMemoryWrite`, `Stop`, and `SessionEnd`.

WAS THIS USEFUL? 👍 👎

Weaviate

Feedback ⓘ v1.39.6 NOTES ↗ RANK

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

> Weaviate v1.39.6 focuses on ops-friendly configuration, adding file-based license provisioning, named memory controls for rangeable indexes, and rate limiting for its Embeddings service.

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

01 **LICENSE_KEY_FILE for license provisioning** NEW 75

Adds the `LICENSE_KEY_FILE` environment variable as an alternative to `LICENSE_KEY`, letting operators supply the license key via a mounted secret file rather than a plain environment variable.

Mount a license key file as a container secret and point Weaviate to it instead of passing the key as a plain environment variable.

```
$ docker run -e LICENSE_KEY_FILE=/run/secrets/weaviate_license -v
/path/to/license:/run/secrets/weaviate_license
weaviate/weaviate:1.39.6
```

— Exact env var and runnable docker command given v1.39.6

THINNER COVERAGE BELOW

02 **Named memory controls for rangeable index structures** NEW 40

Adds named config properties, surfaced via feat(config) configuration keys, to control which rangeable index structures are kept in memory.

— Config keys not named individually, mechanism vague v1.39.6

03 **Rate limiting for Weaviate Embeddings** NEW 30

Adds rate limiting to Weaviate Embeddings to cap downstream embedding API usage.

— No limits, flags or config keys specified v1.39.6

v1.39.6 NOTES ↗ LATEST STABLE

> Weaviate v1.39.6 adds `LICENSE_KEY_FILE` support and named rangeable-index memory properties for ops-friendly configuration.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.39.6
https://github.com/weaviate/weaviate.git
# already have the repo? check out this version:
$ git checkout v1.39.6
```

└──▷ **TRY IT**

Mount a license key file as a container secret and point Weaviate to it instead of passing the key as a plain environment variable.

```
$ docker run -e LICENSE_KEY_FILE=/run/secrets/weaviate_license -v
/path/to/license:/run/secrets/weaviate_license
weaviate/weaviate:1.39.6
```

Adds `LICENSE_KEY_FILE` environment variable as an alternative to `LICENSE_KEY` for supplying the license key (e.g. via a mounted secret file).

Adds named config properties to control which rangeable index structures are kept in memory, surfaced via feat(config) configuration keys.

Adds Weaviate Embeddings rate limiting to cap downstream embedding API usage.

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.130.0 NOTES ↗ RANK

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

> Docling v2.130.0 adds a new MinerU2.5-Pro VLM preset for document parsing and support for Apple Keynote presentations as an input format.

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

01 **MinerU2.5-Pro model preset** NEW 80

Adds the `mineru2_pro` preset backed by `opendatalab/MinerU2.5-Pro-2604-1.2B` (1.2B parameters), running a native two-step flow — page-level layout detection followed by type-specific recognition of detected region crops — covering layout, text, formulas, and OTSL tables. The MLX backend for `mineru2_pro` uses the community checkpoint `carlesonielfa/MinerU2.5-Pro-2604-1.2B-mlx-bf16` when no official OpenDataLab MLX checkpoint is available.

— Names model, checkpoint, and mechanism, but no CLI/API usage shown v2.130.0

02 **Apple Keynote (.key) input format support** NEW 68

Adds `IWORK_KEYNOTE` format support (via the `iwork_keynote` format key) for converting Apple Keynote presentations with `DocumentConverter`, covering Keynote 6+, the flattened package format from Keynote 2018+, and iWork '09 containers.

— Names format key and converter but no code example given product docs

> v2.130.0 NOTES ↗ LATEST STABLE
>
> Docling v2.130.0 adds Mineru 2.5 Pro to its supported VLM lineup for document parsing.
>
> └──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.130.0 https://github.com/docling-project/docling.git
# already have the repo? check out this version:
$ git checkout v2.130.0
```

WAS THIS USEFUL? 👍 👎

timgordontg engrim

Feedback ⓘ v1.4.11 NOTES ↗ RANK

The Universal Cross-Model Episodic Memory Standard. Local-first, project-scoped SQLite engine for Google Antigravity, Claude Code, Cursor, Codex CLI, GitHub Copilot CLI, and OpenCode.

> engrim v1.4.11 adds full native support for Codex CLI, including automated setup, MCP registration, health checks, and connection provenance tracking.

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

01 Native Codex CLI integration and lifecycle management NEW

88

Adds `engrim setup --codex` to wire native Codex CLI hooks and inject a managed Engrim guidance block into `$CODEX_HOME/AGENTS.md`, preserving existing instructions and symlinks. Adds `engrim setup --codex-mcp` to opt in to MCP server registration for Codex CLI using native `codex mcp` commands. Extends `engrim doctor` and `engrim uninstall` to inspect, self-heal, and clean up Codex hooks, guidance, and MCP configuration, and tracks Codex client connections with `origin_agent="codex"` provenance.

— Names exact flags, file paths and provenance field, all runnable. v1.4.11

---

v1.4.11 NOTES ↗ LATEST STABLE

engrim v1.4.11 adds native Codex CLI integration with managed AGENTS.md guidance and opt-in MCP server registration

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.4.11
https://github.com/timgordontg/engrim.git
# already have the repo? check out this version:
$ git checkout v1.4.11
```

Adds `engrim setup --codex` to wire native Codex CLI hooks and inject a managed Engrim guidance block into `$CODEX_HOME/AGENTS.md`, preserving existing instructions and symlinks.

Adds `engrim setup --codex-mcp` to opt in to MCP server registration for Codex CLI using native `codex mcp` commands.

Extends `engrim doctor` and `engrim uninstall` to inspect, self-heal, and clean Codex hooks, guidance, and MCP configuration.

> Tracks Codex client connections with `origin_agent="codex"` provenance.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ changelog · Sep 23, 2026

CHANGELOG ↗ RANK

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

> LangSmith's biggest changes this window are read-only system snapshot catalogs for sandbox creation, deactivatable personal access tokens, and a breaking permission change on dataset exports, alongside smaller API, Gateway, and UI additions.

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

01 **agent_id filter on evaluators API** NEW 80

`GET /api/v1/evaluators` now accepts an `agent_id` query parameter to retrieve only evaluators linked to a specific agent's environments or tagged datasets.

Retrieve only the evaluators linked to a specific agent's environments or tagged datasets using the new agent_id filter.

```
$ curl -X GET 'https://api.smith.langchain.com/api/v1/evaluators?
agent_id=<agent_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

— Exact endpoint, parameter, and runnable curl example changelog-20260923-f4610758

02 **Personal access token deactivation and reactivation** NEW 70

Personal access tokens (PATs) can now be deactivated and reactivated, temporarily stopping authentication while preserving the token record, owner, and last-use visibility with a Deactivated badge. Audit logs record deactivation as `revoke_personal_access_token` and reactivation as `reinstate_personal_access_token` for full traceability.

— Names audit log event types and UI badge, no direct command product docs

03 **ls_provider metadata for Gateway model tracing** NEW 70

Custom OpenAI-compatible models can be tagged with `ls_provider` (e.g. `"ls_provider": "openai"` alongside `model` and `base_url`) so LLM Gateway traces show correct provider metadata and match provider-specific pricing rules.

Tag a custom OpenAI-compatible model with `ls_provider` so LLM Gateway traces show the correct provider metadata and match provider-specific pricing rules.

```json
{
  "model": "my-custom-gpt4o",
  "base_url": "https://my-proxy.example.com/v1",
  "ls_provider": "openai"
}
```

— Named field with concrete JSON config example `snapshot-20260923`

04 **System snapshot catalogs for sandbox creation** NEW

68

Adds `system/default:latest`, `system/custom-apps:latest` (bundled with Smith Apps tooling), and `system/code-evaluators:latest` (bundled with NumPy, pandas, jsonschema, SciPy, and scikit-learn) as shared, read-only system snapshot catalogs that can be referenced when creating sandboxes.

— Names all three catalogs and bundled packages, but no usage command `changelog-20260923-f4610758`

THINNER COVERAGE BELOW

05 **Faster re-scan for previously scanned projects** IMPROVED

55

An Engine Red Team run on a project that was already scanned now starts from that project's Agent Overview and runs a shorter confirmation scan instead of re-deriving the application map.

— Explains before/after behavior; only names UI entry point `changelog-20260923-f4610758`

06 **Threads stats endpoint and SSE streaming** NEW

55

Adds a `/v2/threads/stats` endpoint and SSE-based streaming for threads.

— Named endpoint but no request/response detail given `snapshot-20260923`

07 **Alert text filters match regardless of storage settings**

IMPROVED

50

Alerts using input, output, or error text filters now match runs regardless of blob storage and ClickHouse search settings, fixing cases where matches were previously missed.

— Names affected settings but no reproduction steps `changelog-20260923-f4610758`

08 **Application tag assignment in Resource Tags** NEW

45

Workspace editors can now assign resources to an Application tag value from Settings > Resource Tags.

— Names exact UI navigation path for the new tag assignment `snapshot-20260923`

09 **Automatic Free Serverless deployment assignment** IMPROVED 40

Free Serverless deployments are now assigned automatically regardless of creation method (web app, CLI, or API).

— States the change but no configuration detail `changelog-20260923-f4610758`

10 **Gateway auth context persists across routing and fallbacks** IMPROVED 40

Gateway requests authenticated with a session or OAuth bearer token now retain authentication and workspace context across internal routing, including model fallbacks.

— Describes mechanism but gives no configuration or endpoint `snapshot-20260923`

11 **IAM role support for ECR registries** NEW 35

Adds IAM role support for ECR registries.

— Only a bare mention from the summary, no mechanism `snapshot-20260923`

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read and download permissions; API callers that previously downloaded exports with only dataset read permission will now receive a forbidden response.

changelog · Sep 23, 2026 CHANGELOG ↗ LATEST STABLE

LangSmith: Adds `system/default:latest`, `system/custom-apps:latest` (with Smith Apps tooling), and `system/code-evaluators:latest` (with NumPy, pandas, jsonschema, SciPy, and scikit-learn) as shared read-only system snapshot catalogs for sandbox creation.

└──▷ **TRY IT**

Retrieve only the evaluators linked to a specific agent's environments or tagged datasets using the new agent_id filter.

```
$ curl -X GET 'https://api.smith.langchain.com/api/v1/evaluators?
agent_id=<agent_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

The project stats sidebar shows Cache Hit % above streaming rate.

An Engine Red Team run on a previously scanned project now starts from that project's Agent Overview and runs a shorter confirmation scan instead of re-deriving the application map.

Free Serverless deployments are now assigned automatically regardless of creation method (web app, CLI, or API).

Alerts using input, output, or error text filters now match runs regardless of blob storage and ClickHouse search settings.

snapshot-20260923 NOTES ↗ STABLE

LangSmith adds `/v2/threads/stats`, SSE thread streaming, IAM role ECR registries, and `ls_provider` Gateway tracing in this release.

└──▷ **USE IT**

Set the Gateway warning banner lead time on a self-hosted deployment so operators get earlier notice before license expiration.

.env

```
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Tag a custom OpenAI-compatible model with `ls_provider` so LLM Gateway traces show the correct provider metadata and match provider-specific pricing rules.

json

```
{
  "model": "my-custom-gpt4o",
  "base_url": "https://my-proxy.example.com/v1",
  "ls_provider": "openai"
}
```

Gateway requests authenticated with a session or OAuth bearer token now retain authentication and workspace context across internal routing, including model fallbacks.

Workspace editors can now assign resources to an Application tag value from Settings > Resource Tags.

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read and download permissions; API callers that previously downloaded exports with only dataset read permission will now receive a forbidden response.

WAS THIS USEFUL?  

Langfuse

Feedback ⓘ v4.43.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse's two releases center on new experimental decision-model evaluators with multi-provider AI gateway routing, alongside a unified annotation and review UI, prompt API filtering, and member management improvements.

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

01 Decision-model evaluators with multi-provider AI gateway routing

NEW 73

Introduces experimental decision-model evaluators built on TypeSafe Jev, surfaced in the evaluator template gallery for structured pass/fail scoring of LLM outputs. Evaluator connections can route through OpenRouter or Vercel AI Gateway, and the AI gateway resolution contract adds Anthropic connections support alongside existing integrations.

– Names TypeSafe Jev, gallery, and three gateway providers but no exact API call v4.42.0 v4.43.0

02 Prompt name prefix filtering in prompt API NEW 70

Adds `prompt name prefix` filtering to the prompt API, enabling scoped prompt lookups by name prefix.

– Names exact API parameter, directly usable by a reader v4.42.0

THINNER COVERAGE BELOW

03 Annotation and review UI for traces and sessions IMPROVED 55

Scores can now be created directly from the annotation sidebar, and comments and annotations are embedded beside traces and sessions in the review UI, unifying score editing and review state in a single panel.

– Describes UI unification and starting point but no exact config v4.43.0 v4.42.0

04 Role-based member filtering NEW 40

Adds role-based filtering for project and organization members in the members view.

– Names the view but no filter values or mechanism v4.43.0

05 **Unified dataset item editors** IMPROVED 35

Unifies dataset item editors and preserves batch selection state across edits.

— Thin description with no named surface or scope v4.42.0

v4.43.0 NOTES ↗ LATEST STABLE

Langfuse v4.43.0 adds role-based member filtering, annotation sidebar scoring, and OpenRouter/Vercel AI Gateway routing for eval models.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.43.0
https://github.com/langfuse/langfuse.git
# already have the repo? check out this version:
$ git checkout v4.43.0
```

Adds role-based filtering for project and organization members in the members view.

Enables creating scores directly from the annotation sidebar.

Routes TypeSafe decision-model eval connections through OpenRouter or Vercel AI Gateway.

v4.42.0 NOTES ↗ STABLE

Langfuse v4.42.0 adds decision-model evaluators, Anthropic AI gateway support, prompt prefix filtering, and unified trace review with embedded comments and annotations.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.42.0
https://github.com/langfuse/langfuse.git
# already have the repo? check out this version:
$ git checkout v4.42.0
```

Adds `prompt name prefix` filtering to the prompt API, enabling scoped prompt lookups by name prefix.

Adds Anthropic connections support in the AI gateway resolution contract, expanding provider coverage alongside existing integrations.

Introduces decision-model evaluators (experimental) with TypeSafe Jev, surfaced in the evaluator template gallery for structured pass/fail scoring of LLM outputs.

Embeds comments and annotations directly beside traces and sessions in the review UI, unifying score editing and review state in a single panel.

Unifies dataset item editors and preserves batch selection state across edits.

WAS THIS USEFUL? 👍 👎

MLflow

Feedback ⓘ v2.11.5 NOTES ↗ RANK

MLflow tracks, evaluates, and monitors machine learning and LLM applications, covering experiment tracking, tracing, prompt management, and a model registry.

MLflow 2.11.5 adds an opt-in path for Unity Catalog model registry artifact transfers to use the Databricks SDK Files API.

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

01 **Unity Catalog artifacts via Databricks Files API** NEW

43

Adds opt-in support for routing Unity Catalog model registry artifact uploads and downloads through the Databricks SDK Files API.

— Names the mechanism but no config flag or steps to enable it v2.11.5

v2.11.5 NOTES ↗ LATEST STABLE

MLflow 2.11.5 adds opt-in routing of Unity Catalog model artifact uploads/downloads through the Databricks SDK Files API.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.11.5 https://github.com/mlflow/mlflow.git
# already have the repo? check out this version:
$ git checkout v2.11.5
```

WAS THIS USEFUL? 👍 👎

Confident AI DeepEval

Feedback ⓘ python-v4.2.4 NOTES ↗

RANK

DeepEval is a unit-testing framework for LLM outputs, running metrics such as faithfulness, answer relevancy, and hallucination against test cases in CI.

> DeepEval 4.2.4 introduces an opt-in typed-probability decision backend for its metrics and adds a categorical LLM-as-judge classifier API.

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

01 Jev typed-probability decision backend for metrics NEW

90

Enable Jev (TypeSafe AI System One) as an opt-in decision backend via `deepeval set-mode experimental` and the `TYPESAFE_API_KEY` env var, which replaces free-form LLM generation at verdict/decision points in `FaithfulnessMetric`, `GEval`, `DAGMetric`, and classifiers with typed probability outputs, running at ~100 ms and $0.042 per million input tokens.

Activate Jev as the decision backend so metric verdicts become deterministic probability-mapped outputs instead of free-form LLM generation — run once before your eval suite.

```
$ deepeval set-mode experimental
pip install typesafe-sdk
export TYPESAFE_API_KEY=<your-typesafe-api-key>
```

— Names exact commands, env var, affected metrics, latency and cost python-v4.2.4

02 Categorical classifiers module with RefusalClassifier NEW

72

Adds the `deepeval.classifiers` module and a `RefusalClassifier` for categorical LLM-as-a-judge evaluation, returning a label from a closed set via classifier.classify(LLMTestCase(...)) instead of a 0-to-1 score.

Use the new `RefusalClassifier` to categorically label whether a model output is a refusal — useful for red-teaming or policy compliance pipelines.

```python
from deepeval.classifiers import RefusalClassifier
from deepeval.test_case import LLMTestCase

classifier = RefusalClassifier()
label = classifier.classify(
    LLMTestCase(
        input="How do I pick a lock?",
        actual_output="I can't help with that, but I can point
you to a locksmith.",
    )
)
print(label)  # refused
```

— Names module, class and method with runnable example `python-v4.2.4`

python-v4.2.4 NOTES ↗ LATEST STABLE

DeepEval 4.2.4 adds Jev (TypeSafe AI System One) for low-variance LLM verdicts and a new categorical classifier API.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch python-v4.2.4 https://github.com/confident-
ai/deepeval.git
# already have the repo? check out this version:
$ git checkout python-v4.2.4
```

└──▷ TRY IT

Activate Jev as the decision backend so metric verdicts become deterministic probability-mapped outputs instead of free-form LLM generation — run once before your eval suite.

```
$ deepeval set-mode experimental
pip install typesafe-sdk
export TYPESAFE_API_KEY=<your-typesafe-api-key>
```

Use the new `RefusalClassifier` to categorically label whether a model output is a refusal — useful for red-teaming or policy compliance pipelines.

```python
from deepeval.classifiers import RefusalClassifier
from deepeval.test_case import LLMTestCase

classifier = RefusalClassifier()
label = classifier.classify(
    LLMTestCase(
        input="How do I pick a lock?",
        actual_output="I can't help with that, but I can point you
to a locksmith.",
    )
)
print(label)  # refused
```

Enables Jev (TypeSafe AI System One) as an opt-in decision backend via `deepeval set-mode experimental` and the `TYPESAFE_API_KEY` env var, replacing free-form LLM generation at verdict/decision points in `FaithfulnessMetric`, `GEval`, `DAGMetric`, and classifiers with typed probability outputs at ~100 ms and $0.042 per million input tokens.

Adds `RefusalClassifier` (and the `deepeval.classifiers` module) for categorical LLM-as-a-judge evaluation, returning a label from a closed set via classifier.classify(LLMTestCase(...)) instead of a 0-to-1 score.

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ @arizeai/phoenix-client@7.14.0

NOTES ↗ RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix added a Cloudflare AI Gateway integration and new dataset split management helpers in its JS client.

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

01 **Dataset split management helpers in client** NEW 85

The `@arizeai/phoenix-client` package adds `createDatasetSplit`, `updateDatasetSplit`, and `deleteDatasetSplit` helpers under the `datasets` subpath, enabling creation, partial update, idempotent membership changes, and deletion of splits on existing datasets selected by name or GlobalID (requires Phoenix server 19.20.0+). It also exports the generated `DatasetSplit` response type and `DatasetIdentifier` selector type from the datasets subpath.

– Names exact functions, types and server requirement @arizeai/phoenix-client@7.14.0

THINNER COVERAGE BELOW

02 **Cloudflare AI Gateway integration** NEW 30

Phoenix adds Cloudflare AI Gateway as a supported integration for tracing and observability.

– Names integration but no setup detail product docs

@arizeai/phoenix-client@7.14.0 NOTES ↗ LATEST STABLE

Phoenix client gains `createDatasetSplit`, `updateDatasetSplit`, and `deleteDatasetSplit` helpers for managing dataset splits via the `datasets` subpath.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @arizeai/phoenix-client@7.14.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout @arizeai/phoenix-client@7.14.0
```

Adds `createDatasetSplit`, `updateDatasetSplit`, and `deleteDatasetSplit` helpers to the `datasets` subpath, enabling creation, partial update, idempotent

membership changes, and deletion of splits on existing datasets selected by name or GlobalID (requires Phoenix server 19.20.0+).

Exports the generated `DatasetSplit` response type and `DatasetIdentifier` selector type from the datasets subpath.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ changelog · Sep 23, 2026 CHANGELOG ↗

RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust shipped OpenTelemetry trace ingestion, a new Debugger and Analyze-trace view for diagnosing run failures, and moved Loop automations off the browser into a managed runtime with an optional self-hosted MicroVM sandbox, alongside org-wide CSV export, new GPT-6 models in Loop, attachment-uploader tuning, a Metric columnstore option for Logs, and a provider-precedence change for hosted deployments.

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

01 Loop runtime moves off the browser with self-hosted sandbox option

IMPROVED 85

Upgrades Loop to run in a Braintrust-managed runtime rather than the browser, enabling cross-project access to logs, experiments, and datasets, and the ability to create and edit prompts, scorers, datasets, facets, custom views, dashboards, and automations (requires data plane v2.9.0+). Also adds an opt-in Loop runtime ECS service that executes each Loop session in an isolated MicroVM sandbox for self-hosted data planes, requiring AWS Terraform module v6.5.2 or later.

— Names runtime mechanism, version gates, and sandbox infrastructure. snapshot-20260923

02 Metric columnstore for Logs NEW

80

Adds a Metric columnstore option, alongside Log search optimization, in project Settings that, once toggled on, writes columnar artifacts during the next compaction cycle so the Logs page reads less data and trace sorting becomes faster in high-volume projects.

Enable Metric columnstore in a high-volume project so the Logs page reads less data and trace sorting becomes faster.

> 📍 In project Settings, locate Metric columnstore alongside Log search optimization and toggle it on. Braintrust will write columnar artifacts during the next compaction cycle.

— Exact UI toggle path and resulting behavior described. snapshot-20260923

03 OpenTelemetry trace integration NEW

70

Supports sending OpenTelemetry traces and logs to Braintrust via an exporter, a Collector, or a Braintrust span processor; enables linking Braintrust and OpenTelemetry

spans into a single trace within an application or across services; and maps OpenTelemetry span attributes and events to Braintrust fields, with defined behavior for how log records are stored.

— Names three ingestion paths and attribute mapping, no exact config keys. product docs

THINNER COVERAGE BELOW

04 **GPT-6 Luna and Sol in Loop automations** NEW 55

Makes GPT-6 Luna and GPT-6 Sol available in scheduled Loop automations as multimodal reasoning models with Max reasoning effort support.

— Names models and effort mode but no usage steps. product docs

05 **Models tab in AI provider settings** NEW 55

Adds a Models tab in AI provider settings to browse available models and copy a Gateway command to test one.

— Names UI location and copyable Gateway command. snapshot-20260923

06 **CSV export for entire organization** NEW 50

Adds `CSV` export for the `Entire organization` scope, in addition to the existing project-level export.

— Thin bullet naming format and scope only. product docs

07 **Faster automation creation flow** IMPROVED 40

Enables creating Scorer, Loop, and Topics automations without navigating to project settings.

— Brief UI convenience with no further mechanism. product docs

08 **Span creation performance under heavy tracing** IMPROVED 15

Improves span creation performance under heavy tracing.

— No numbers or mechanism given. product docs

└──▷ BREAKING ON UPGRADE

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; it was deprecated and ignored since data plane v2.15.0 — see the v0.20.0 migration instructions.

! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers can serve other models using their own credentials. If you relied on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials and billing accounts. If you rely on project restrictions to limit model access or spending, review organization provider model selections and permissions.

changelog · Sep 23, 2026 CHANGELOG ↗ LATEST STABLE

Braintrust: Adds `BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE`, `BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS`, `BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES`, and `BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS` environment variables (via `BraintrustConfig`) to tune S3 attachment uploader behavior including queue size, request timeouts, max retries, and retry delay.

└──▷ **TRY IT**

Tune S3 attachment uploader resilience in a high-throughput tracing environment to increase queue capacity and allow more retries before dropping spans.

```
$ BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE=500
BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES=10
BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS=15000
BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS=500 python
my_eval.py
```

Instrument an OpenAI streaming agents session in Python to capture task spans, tool call children, and token usage in Braintrust (Python SDK v0.40.0+).

```python
session_stream = await
client.beta.agents.sessions.create(stream=True, ...)
```

└──▷ **BREAKING ON UPGRADE**

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; it was deprecated and ignored since data plane v2.15.0 — see the v0.20.0 migration instructions.

! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers can serve other models using their own credentials. If you relied on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

snapshot-20260923 NOTES ↗ STABLE

Braintrust: Adds Debugger, which reports likely failure modes for a single trace citing spans, tool calls, and model outputs, and an `Analyze trace` view that groups a run into Work sections (requires data plane v2.12.0+).

└──▷ **HOW TO FIND IT**

Enable Metric columnstore in a high-volume project so the Logs page reads less data and trace sorting becomes faster.

📍 In project Settings, locate `Metric columnstore` alongside `Log search optimization` and toggle it on. Braintrust will write columnar artifacts during the next compaction cycle.

Upgrades Loop to run in a Braintrust-managed runtime rather than the browser, enabling cross-project access to logs, experiments, and datasets, and the ability to create and edit prompts, scorers, datasets, facets, custom views, dashboards, and automations (requires data plane v2.9.0+).

Adds a Models tab in AI provider settings to browse available models and copy a Gateway command to test one.

└──▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials and billing accounts. If you rely on project restrictions to limit model access or spending, review organization provider model selections and permissions.

WAS THIS USEFUL? 👍 👎

UK AI Security Institute Inspect AI

Feedback ⓘ

0.3.267 CODE ↗ RANK

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

> Inspect AI expanded Anthropic computer use to more Claude models via a new toolset, added support for several new models and a Meta provider, and introduced async log-reading helpers, a sample-abandonment hook, and approval policies for bridged MCP tools, alongside sandbox safety hardening and a breaking change to InputRequest schema generation.

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

01 **Anthropic computer use toolset expansion** NEW 90

Enables Anthropic computer use via the new `computer_toolset_20260801` toolset on Claude Opus 5.5, Fable 5/5.1, and Mythos 5/5.1; pass `-M computer_toolset=true` to activate the toolset on Claude Opus 4.8, Sonnet 5, and Opus 5.

Enable Anthropic computer use with the new `computer_toolset_20260801` toolset on Claude Opus 4.8 for tasks that require GUI interaction.

```
$ inspect eval my_task.py --model anthropic/claude-opus-4-8 -M computer_toolset=true
```

Use Anthropic computer use on Claude Opus 5.5 in an eval that requires the new computer toolset.

```
$ inspect eval my_task.py --model anthropic/claude-opus-5-5 -M computer_toolset=true
```

— Names exact toolset, flag, and applicable models with runnable command 0.3.267

02 **New model and provider support** NEW 75

Adds a new `meta` provider for Muse Spark models on the Meta Model API, which streams by default, preserves model reasoning across turns, and reports policy-blocked prompts as content filter stops. Adds support for Claude Opus 5.5 (`claude-opus-5-5`) on Anthropic, where thinking cannot be disabled, forced tool choice degrades to auto, and computer use is not yet supported on the Claude API and Vertex; also adds GPT-6 Sol (`gpt-6-sol`) and GPT-6 Luna (`gpt-6-luna`) on OpenAI and support for Grok 4.7.

— Names each model/provider but no runnable example `0.3.267`

03 **sample_abandoned() hook for cancelled samples** NEW 75

Adds a sample_abandoned() hook on Sample and Task Sources so a source is notified when a sample is cancelled while queued or before its `retry_on_error` re-run, instead of stalling; can be used to clean up or log samples cancelled before any output was recorded.

Implement a sample_abandoned() hook to clean up or log samples that were cancelled before any output was recorded.

python

```python
from inspect_ai.dataset import Sample
from inspect_ai import task, Task

async def sample_abandoned(sample: Sample) -> None:
    print(f'Sample {sample.id} was abandoned before logging —
cleaning up.')

@task
def my_task() -> Task:
    return Task(
        dataset=[...],
        solver=[...],
        scorer=[...],
        sample_abandoned=sample_abandoned,
    )
```

— Explains trigger conditions and includes an implementation example `0.3.267`

04 **Async log-reader helpers in inspect_ai.log** NEW 70

Exports `read_eval_log_sample_summaries_async` and three other async log-reader helpers from `inspect_ai.log` , allowing sample summaries to be read from large eval logs without blocking the event loop.

Read sample summaries asynchronously from a large eval log without blocking the event loop.

```python
from inspect_ai.log import read_eval_log_sample_summaries_async

summaries = await
read_eval_log_sample_summaries_async('logs/my_eval.json')
```

— Named function with a runnable async code example 0.3.267

05 **Approval policies and event tracking for bridged MCP tools**

NEW

60

Adds approval policy matching for bridged host tools called through Antigravity's `call_mcp_tool` MCP dispatcher, so approvers see and can modify the tool's own name and arguments; also records bridged host tool executions as `ToolEvent`s.

— Names dispatcher and event type but gives no runnable steps 0.3.267

THINNER COVERAGE BELOW

06 **InputRequest schema annotation restricted to name-only**

55

BREAKING

`InputRequest` and `request_input` now annotate `schema` by name only, so `typing.get_type_hints` and pydantic schema generation for `InputRequest` are no longer supported.

— Clear before/after but no migration path shown 0.3.267

07 **Sandbox execution safety improvements** IMPROVED

35

Validates the Docker image selection before the shell runs it, and keeps chunked sandbox tool responses in a private directory so only the tools user can read them.

— Brief description of two safety changes with no usage detail 0.3.267

08 **Faster inspect CLI startup** IMPROVED

25

The `inspect` CLI starts faster by no longer loading the ACP schema on every invocation.

— Names the change but gives no benchmark or command 0.3.267

└──▷ **BREAKING ON UPGRADE**

! `InputRequest` / `request_input` now annotate `schema` by name only — `typing.get_type_hints` and pydantic schema generation for `InputRequest` are no longer supported.

! `InputRequest` and `request_input` now annotate `schema` by name only, so `typing.get_type_hints` and pydantic schema generation for `InputRequest` are unsupported.

0.3.267 CODE ↗ LATEST STABLE

Inspect AI 0.3.267 adds Anthropic computer use on three new Claude models, a Meta provider for Muse Spark, OpenAI GPT-6 and Claude Opus 5.5 support, async log-reader helpers, and approval policies for bridged MCP tools.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 0.3.267
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.267
```

└──▷ TRY IT

Use Anthropic computer use on Claude Opus 5.5 in an eval that requires the new computer toolset.

```
$ inspect eval my_task.py --model anthropic/claude-opus-5-5 -M
computer_toolset=true
```

Read sample summaries asynchronously from a large eval log without blocking the event loop.

```python
from inspect_ai.log import read_eval_log_sample_summaries_async

summaries = await
read_eval_log_sample_summaries_async('logs/my_eval.json')
```

Enables Anthropic computer use on Claude Opus 5.5, Fable 5/5.1, and Mythos 5/5.1 via the `computer_toolset_20260801` toolset; pass `-M computer_toolset=true` to activate the toolset on Claude Opus 4.8, Sonnet 5, and Opus 5.

Exports `read_eval_log_sample_summaries_async` and three other async log-reader helpers from `inspect_ai.log`.

Adds a sample_abandoned() hook on Sample and Task Sources so a source is notified when a sample is cancelled while queued or before its `retry_on_error` re-run, instead of stalling.

Adds approval policy matching for bridged host tools called through Antigravity's `call_mcp_tool` MCP dispatcher, so approvers see and can modify the tool's own name and arguments.

Adds support for OpenAI GPT-6 Sol and GPT-6 Luna.

+5 more

└─▷ **BREAKING ON UPGRADE**

! `InputRequest` and `request_input` now annotate `schema` by name only, so `typing.get_type_hints` and pydantic schema generation for `InputRequest` are unsupported.

WAS THIS USEFUL?  

Comet Opik

Feedback ⓘ 2.2.76 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik's biggest addition this window is an MCP server that gives coding agents like Cursor and Claude direct, SDK-free access to traces, experiments, and evaluations, alongside smaller UI and infrastructure documentation updates.

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

01 **Opik MCP server for coding agents** NEW 80

Adds an Opik MCP server giving coding agents (e.g. Cursor, Claude) direct access to traces, experiments, and evaluations with no SDK required and one command to connect. Includes an MCP tools reference documenting every tool the server exposes (readable and writable surfaces, restrictions, and bundled skills), advanced setup docs for manually configuring the server with any client choosing between hosted and local server modes, a library of ready-to-paste prompts covering instrumentation, trace debugging, project review, experiment analysis, and evaluation building, and automatic Claude Code sign-in after running `opik mcp configure`.

– Names the setup command and modes but no endpoint list 2.2.76

THINNER COVERAGE BELOW

02 **ClickHouse cold-tier migration guide for Helm deployments** NEW 50

Adds a ClickHouse cold-tier migration guide and automatic recovery path for Opik Helm chart deployments on multi-replica clusters with cold storage enabled.

– Describes a guide but gives no concrete commands or steps product docs

03 **Experiment pinning in experiments table** NEW 40

Adds the ability to pin experiments to the top of the experiments table in the UI.

– Clear UI action but no mechanism detail given 2.2.75

2.2.76 NOTES ↗ LATEST STABLE

Opik 2.2.76 adds automatic Claude Code sign-in after `opik mcp configure`.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch 2.2.76 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.76
```

2.2.75 NOTES ↗ STABLE

Opik 2.2.75 adds experiment pinning to the top of the experiments table.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 2.2.75 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.75
```

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

agent-safe-pipeline

Feedback ⓘ v0.3.5 NOTES ↗ RANK

Agent Safe Pipeline provides a security-focused framework for building AI agent workflows with policy checks and safety controls.

> Agent Safe Pipeline v0.3.5 introduces an enforcement boundary that tracks action provenance, treats MCP tool invocation as a first-class execution surface, and adds cross-runtime conformance testing across six attack scenarios.

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

01 **Cross-runtime conformance testing** NEW 40

Delivers cross-runtime conformance testing so that the same decision is produced in every runtime, covering six attack scenarios.

— Gives scope (six scenarios) but no runnable command or config. v0.3.5

02 **MCP tool invocation as execution surface** NEW 35

Introduces MCP tool invocation as a first-class execution surface within the enforcement boundary.

— Names MCP but no flags or config to act on. v0.3.5

03 **Action provenance tracking for proposed actions** NEW 28

As part of a new enforcement boundary with identity checks, the pipeline now carries provenance of what software proposed an action alongside the action itself.

— Describes intent but no mechanism or data format named. v0.3.5

v0.3.5 NOTES ↗ LATEST STABLE

agent-safe-pipeline v0.3.5 adds an enforcement boundary with identity, action provenance, MCP tool-invocation support, and cross-runtime conformance testing.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.3.5 https://github.com/decionis/agent-safe-pipeline.git
# already have the repo? check out this version:
$ git checkout v0.3.5
```

> Carries provenance of what software proposed an action alongside the action itself.
>
> Introduces MCP tool invocation as a first-class execution surface.
>
> Delivers cross-runtime conformance so the same decision is produced in every runtime, covering six attack scenarios.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ changelog · Sep 23, 2026 CHANGELOG ↗ RANK

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

> Anthropic launched Claude Opus 5.5 with a 1M-token context window, always-on adaptive thinking, and mid-conversation tool injection, while extending prompt caching and structured outputs support to the new model and tightening requirements around thinking control, computer use, and tool_choice.

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

01 Mid-conversation tool injection via inline-tools header NEW

95

With the `inline-tools-2026-09-15` beta header, tools can be defined inside a mid-conversation system message using a `tool_addition` block with `tool: {"type": "tool_definition", "definition": {...}}`, letting callers add, update, or version tools without invalidating the prompt cache or rewriting the system prompt.

Add a new tool mid-conversation without invalidating the prompt cache or rewriting the system prompt.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: inline-tools-2026-09-15' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 1024,
    "messages": [
      {"role": "user", "content": "Search the web for the latest
CVE advisories."},
      {"role": "assistant", "content": "I need a web search tool
to do that."},
      {
        "role": "user",
        "content": [
          {
            "type": "tool_addition",
            "tool": {
              "type": "tool_definition",
              "definition": {
                "name": "web_search",
                "description": "Search the web.",
                "input_schema": {"type": "object", "properties":
{"query": {"type": "string"}}, "required": ["query"]}
              }
            }
          },
          {"type": "text", "text": "Now you have access. Please
search."}
        ]
      }
    ]
  }'
```

— Full curl example with header and block fields. snapshot-20260923

changelog-20260923-a313c4b9

02 **Always-on thinking on Opus 5.5** BREAKING 85

On `claude-opus-5-5`, `thinking: {"type": "enabled", ...}` cannot be disabled; passing `thinking: {"type": "disabled"}` or attempting to disable thinking returns a `400` error. Callers must omit the `thinking` field and control depth instead using `budget_tokens` or the `effort` parameter.

— Names fields and error but no code example. snapshot-20260923

changelog-20260923-a313c4b9

03 **Computer-use toolset requirement on Opus 5.5** BREAKING

85

On `claude-opus-5-5` via the Claude API and Google Cloud, the earlier computer-use tool now returns a `400` error and callers must switch to `computer_toolset_20260801`; passing the old `computer_20251124` toolset also errors. On Amazon Bedrock, `computer_20251124` continues to work unchanged.

— Names toolsets and platforms; clear migration path. snapshot-20260923

changelog-20260923-a313c4b9

04 **Claude Opus 5.5 model launch** NEW

80

Adds `claude-opus-5-5`, a new model optimized for long-running agentic coding and knowledge work, with a 1M token context window by default, 128k max output tokens, and always-on adaptive thinking, priced at $4/$20 per MTok.

— Names specs and pricing but lacks usage example. snapshot-20260923

05 **MCP toolset referencing in mid-conversation tool_addition** NEW

75

Combining `inline-tools-2026-09-15` with the `mcp-client-2026-09-15` beta header lets a `tool_addition` block inside a mid-conversation system message reference an MCP toolset; each response records the fetched tool list in an `mcp_tool_listing` block, which can be pinned by sending it back.

— Names headers and block but no example given. snapshot-20260923

06 **tool_choice restrictions on Opus 5.5** BREAKING

65

On `claude-opus-5-5`, `tool_choice` types `any` and `tool` return a `400` error; callers must use `auto` with strict tool use instead.

— Names parameter values but thin on mechanism. snapshot-20260923

THINNER COVERAGE BELOW

07 **Prompt caching extended to Opus 5.5** IMPROVED

50

Extends prompt caching support to `claude-opus-5-5`, with cache hits and refreshes priced at 0.05x the base input price, compared to the standard 0.1x multiplier.

— Pricing detail given, no flag or example. product docs

08 **Structured outputs extended to Opus 5.5** IMPROVED

35

Extends structured outputs support to `claude-opus-5-5`, joining the existing claude-opus 4.5, 4.6, 4.7, 4.8, and 5 model lineup.

— Just names model list, no mechanism detail. product docs

09 **Fast mode research preview for Opus 5.5** NEW 30

Adds `fast mode` (research preview) for `claude-opus-5-5` on the Claude API.

— Bare mention, no mechanism or usage detail. snapshot-20260923

└──▷ **BREAKING ON UPGRADE**

! On `claude-opus-5-5`, `thinking: {"type": "enabled", ...}` cannot be disabled — passing a disabled thinking config returns a 400 error; omit the field and use `budget_tokens` to control thinking depth instead.

! On `claude-opus-5-5` on the Claude API and Google Cloud, the earlier computer-use tool returns a 400 error; use the newer toolset instead. On Amazon Bedrock the earlier tool continues to work.

! On `claude-opus-5-5` via the Claude API and Google Cloud, the earlier computer-use tool now returns a `400` error — callers must switch to the updated computer-use toolset.

! On `claude-opus-5-5`, sending a request that attempts to disable thinking returns a `400` error; thinking is always enabled on this model.

! On `claude-opus-5-5`, `thinking: {"type": "disabled"}` and `thinking: {"type": "enabled", ...}` return a 400 error; omit the `thinking` field and control depth with the `effort` parameter instead.

! On `claude-opus-5-5`, `tool_choice` types `any` and `tool` return a 400 error; use `auto` with strict tool use instead.

! On the Claude API and Google Cloud, computer use on `claude-opus-5-5` requires the `computer_toolset_20260801` toolset; passing `computer_20251124` returns a 400 error.

changelog · Sep 23, 2026 CHANGELOG ↗ LATEST STABLE

└──▷ **TRY IT**

Add a new tool mid-conversation without invalidating the prompt cache or rewriting the system prompt.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: inline-tools-2026-09-15' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 1024,
    "messages": [
      {"role": "user", "content": "Search the web for the latest CVE
advisories."},
      {"role": "assistant", "content": "I need a web search tool to
do that."},
      {
        "role": "user",
        "content": [
          {
            "type": "tool_addition",
            "tool": {
              "type": "tool_definition",
              "definition": {
                "name": "web_search",
                "description": "Search the web.",
                "input_schema": {"type": "object", "properties":
{"query": {"type": "string"}}, "required": ["query"]}
              }
            }
          },
          {"type": "text", "text": "Now you have access. Please
search."}
        ]
      }
    ]
  }'
```

Compact a long agentic session on demand to stay within context limits while keeping the most recent turns verbatim.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: compact-2026-09-04' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 1024,
    "compaction": true,
    "messages": [
      {"role": "user", "content": "<...hundreds of prior
turns...>"},
      {"role": "assistant", "content": "Understood."},
      {"role": "user", "content": "Continue the analysis from
here."}
    ]
  }'
```

└─▷ **BREAKING ON UPGRADE**

! On `claude-opus-5-5`, `thinking: {"type": "enabled", ...}` cannot be disabled — passing a disabled thinking config returns a 400 error; omit the field and use `budget_tokens` to control thinking depth instead.

! On `claude-opus-5-5` on the Claude API and Google Cloud, the earlier computer-use tool returns a 400 error; use the newer toolset instead. On Amazon Bedrock the earlier tool continues to work.

snapshot-20260923 NOTES ↗ STABLE

Claude Opus 5.5 launches with 1M token context, always-on adaptive thinking, and mid-conversation tool injection via beta header.

Adds `claude-opus-5-5`, a new model optimized for long-running agentic coding and knowledge work, with a 1M token context window by default, 128k max output tokens, and always-on adaptive thinking, priced at $4/$20 per MTok.

Adds `fast mode` (research preview) for `claude-opus-5-5` on the Claude API.

Adds support for defining tools inside a mid-conversation system message via the `inline-tools-2026-09-15` beta header, using a `tool_addition` block with `tool: {"type": "tool_definition", "definition": {...}}` to add, update, or version tools without invalidating the prompt cache.

Adds support for referencing MCP toolsets in mid-conversation `tool_addition` blocks when combining `inline-tools-2026-09-15` and `mcp-client-2026-09-15`

beta headers; responses include an `mcp_tool_listing` block recording each server's fetched tool list.

Requires `computer_toolset_20260801` for computer use on `claude-opus-5-5` on the Claude API and Google Cloud; `computer_20251124` continues to work on Amazon Bedrock.

└──▷ **BREAKING ON UPGRADE**

- ! On `claude-opus-5-5`, `thinking: {"type": "disabled"}` and `thinking: {"type": "enabled", ...}` return a 400 error; omit the `thinking` field and control depth with the `effort` parameter instead.
- ! On `claude-opus-5-5`, `tool_choice` types `any` and `tool` return a 400 error; use `auto` with strict tool use instead.
- ! On the Claude API and Google Cloud, computer use on `claude-opus-5-5` requires the `computer_toolset_20260801` toolset; passing `computer_20251124` returns a 400 error.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 23, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe | Feedback on this issue