THE AI TOOLCHAIN

NO. 004

Head

THE WEEK'S TOP PICKS

PUBLISHED SEPTEMBER 11, 2026 · FRIDAYS

// SHARE THIS WEEK ▽ Top 5 ▽ Top Pick

The best tooling updates that shipped this week.

// HOW THESE PICKS ARE MADE

Every feature release from the 354 tools on our watchlist goes into the daily newsletter. Once a week we read the whole field side by side and choose our top picks, judged on two questions: **how deep and complete is the single best capability in the release**, and **how much it changes what you can actually do**.

A tool is judged on everything it shipped that week, so a project that releases daily gets credit for the sum — and still only takes one slot. We would rather run a short list than a padded one.

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

What stands out in this week's picks, grouped by what it lets you do. Every tool named links to its pick below.

Three unrelated things stand out today: Cursor can coordinate long-running coding work across many agents and your own machines; Pinecone now mixes BM25 with vectors in one index; Braintrust starts finding recurring trace failures without a human spelunking dashboards first.

BUILD

Hand a coding project to a coordinator instead of a single chat loop

Projects gives teams a persistent agent that can split work across subagents while staying subscribed to Slack, schedules, and PRs. Self-hosted machine pools also let cloud agents run on your infrastructure, so longer coding jobs can use controlled compute instead of a vendor-only runtime.

Cursor

BUILD

Run lexical and semantic retrieval from the same index

BM25 is now GA inside Pinecone's vector index, so teams can combine exact-term matching with embedding similarity without maintaining a separate search stack. The new admin and access-control APIs also make it easier to automate org, project, API-key, service-account, and role-binding setup instead of wiring them by hand.

Pinecone

OPERATE

Let trace investigation run before an engineer opens the dashboard

Patterns runs Loop on a schedule to surface recurring production issues, while Debugger diagnoses individual traces. That turns trace review from a manual hunt through logs into a standing background process that points teams at repeated failures and likely causes.

Braintrust

DEPLOY

Push more serving work onto the hardware and runtime path you already picked

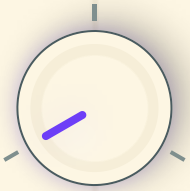
vLLM added request admission-control flags and Tenstorrent serving support; SGLang added cross-node MoE decoding, quantized MoE optimizations, and more speculative-decoding backends; llama.cpp improved memory-efficient loading, distributed inference, and KV-cache behavior. Together, these make capacity, latency, and hardware fit less dependent on moving to a different serving stack.

vLLM · SGLang · llama.cpp

VIEW

ISSUE VIEW

full issue



FULL ISSUE | EXPAND IN PLACE | PREVIEW PANE

Do you prefer **full issue**? 👍 👎

// THE PICKS

01	**Cursor**	AI Coding Agents
02	**Pinecone**	Vector DBs & RAG
03	**Anthropic**	Frontier Models
04	**Braintrust**	AI Observability & Evals
05	**vLLM**	Local LLM Runtimes

DENSITY BRIEFED **EVERYTHING** Every change in this week's picks, in full.

all 77 changes

01 Cursor

4 RELEASES · 2026-09-02 → 2026-09-11

AI Coding Agents

Cursor is an AI-powered code editor built on VS Code that uses large language models to assist with writing, debugging, and refactoring code.

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 5/5

Projects launches a coordinator agent that plans and delegates work across thousands of parallel cloud subagents and keeps running after you close your laptop, with shared context files that sync across every agent and Subscriptions that watch a Slack channel or schedule to act autonomously on new signals.

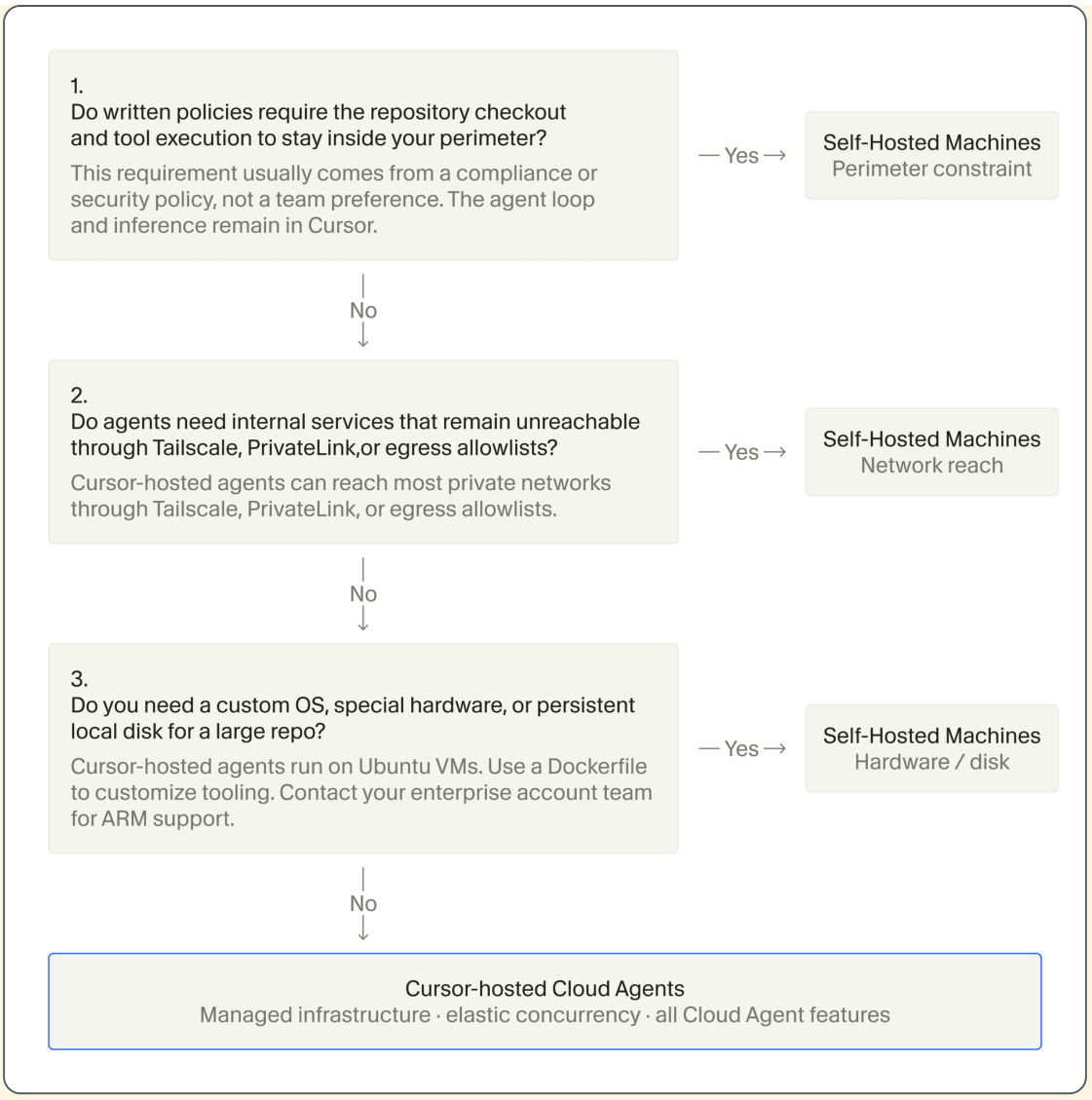
> Cursor's biggest window in a while: Projects brings a persistent coordinator agent that delegates to thousands of subagents with shared context and Slack/schedule/PR subscriptions, alongside self-hosted machine pools for running cloud agents on your own infrastructure, a new Origin code hosting service, and a batch of smaller cloud-agent workflow additions (`/goal`, Custom Modes, Builds, Publish, live port-forwarding).

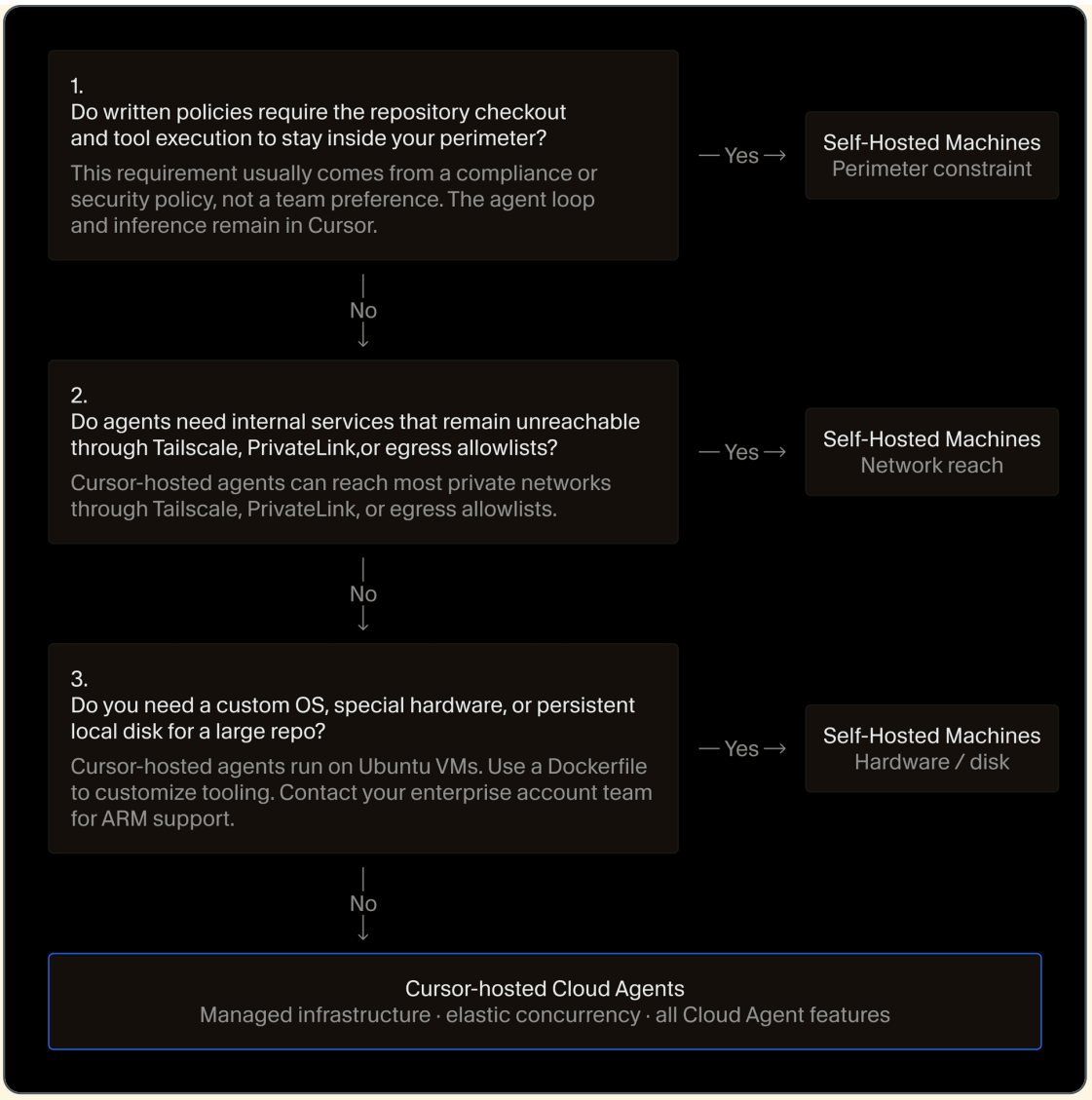
└──▷ WHAT SHIPPED · 12 FEATURES most completely described first ? what's the number?

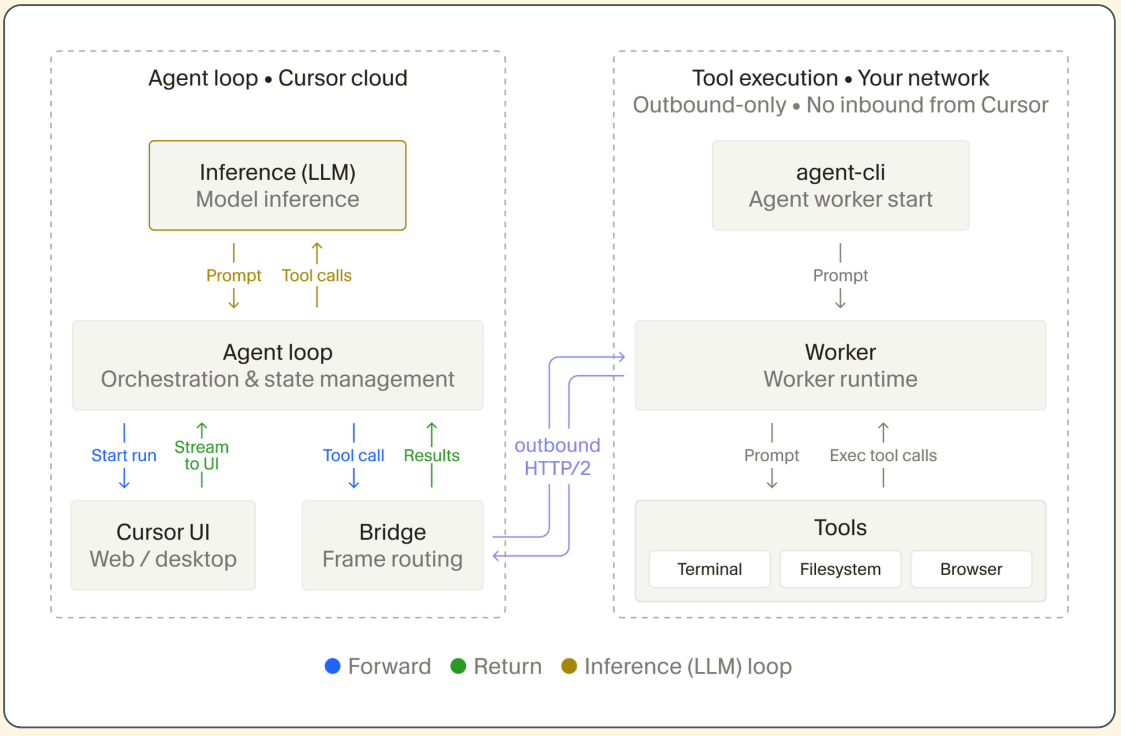
01 Self-hosted machines and worker pools for cloud agents NEW

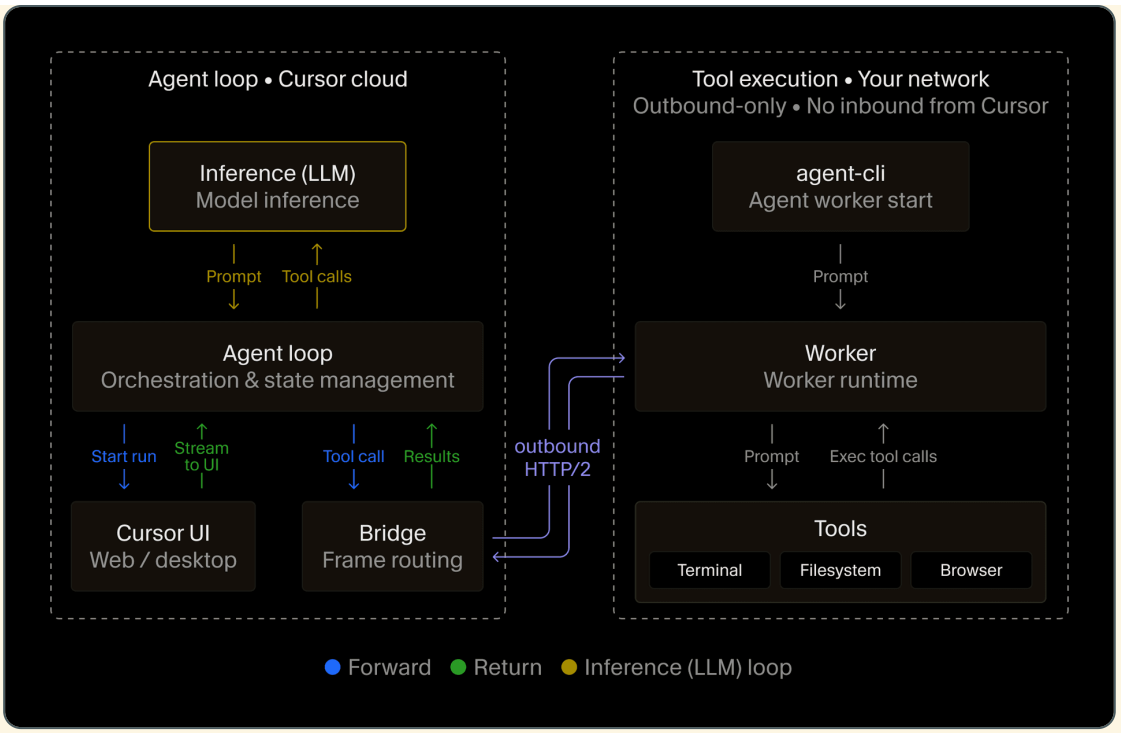
100

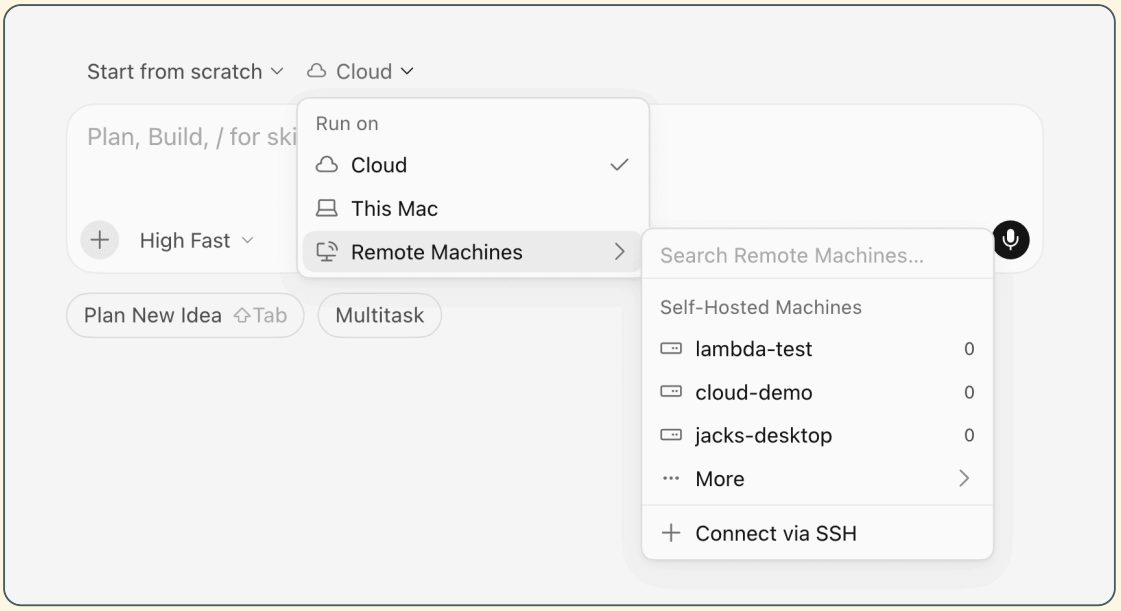
The `cursor agent worker start` CLI subcommand registers a machine as a self-hosted worker, opening a long-lived outbound HTTPS connection to the Cursor cloud so tool execution, codebase, build outputs, and secrets stay entirely within your own network. Workers are organized via 'My Machines' (single laptop/VM) or Pools/Team Pools (named queues of workers) that auto-scale and hibernate idle machines, and can run on pre-existing sandbox infrastructure including AWS Lambda, Coder, Cloudflare, Daytona, Modal, Namespace, Vercel, and E2B, or on custom hardware such as GPUs, Apple silicon Macs (for iOS/macOS builds), and Kubernetes. Self-hosted workers on Linux and Mac support computer use — click, type, screenshot, and browser control via Chrome or Chromium — with live desktop viewing and takeover from Cursor.

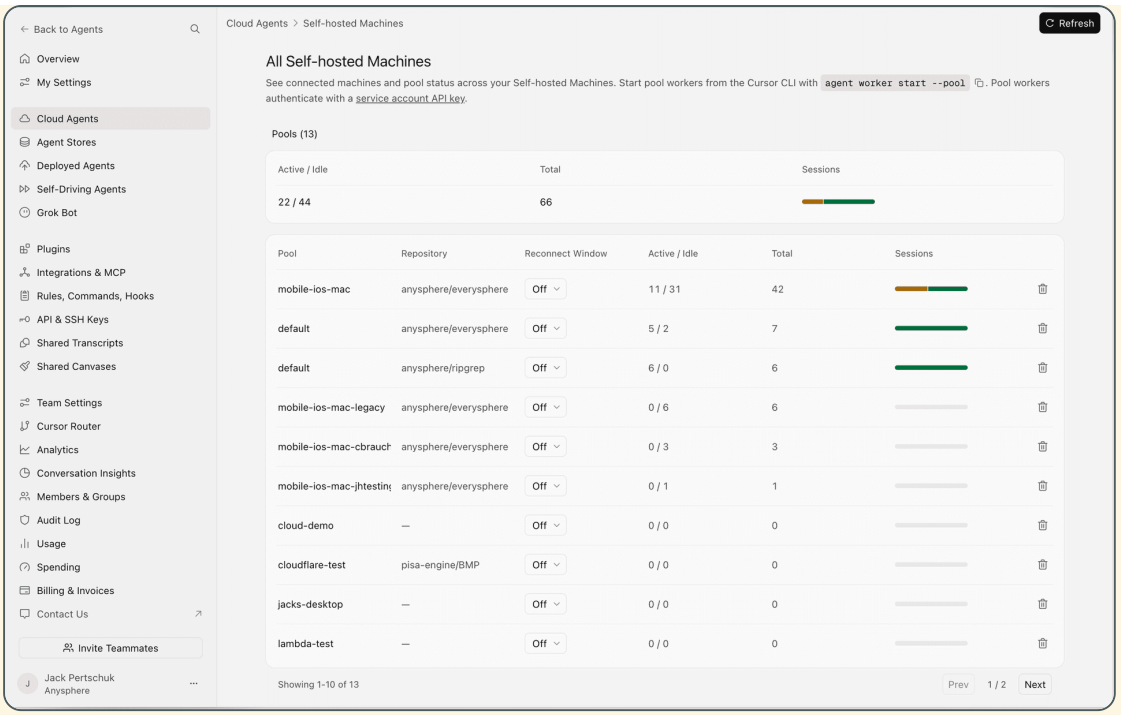

1.
Do written policies require the repository checkout and tool execution to stay inside your perimeter?
This requirement usually comes from a compliance or security policy, not a team preference. The agent loop and inference remain in Cursor.
— Yes →
Self-Hosted Machines
Perimeter constraint
No
2.
Do agents need internal services that remain unreachable through Tailscale, PrivateLink,or egress allowlists?
Cursor-hosted agents can reach most private networks through Tailscale, PrivateLink, or egress allowlists.
— Yes →
Self-Hosted Machines
Network reach
No
3.
Do you need a custom OS, special hardware, or persistent local disk for a large repo?
Cursor-hosted agents run on Ubuntu VMs. Use a Dockerfile to customize tooling. Contact your enterprise account team for ARM support.
— Yes →
Self-Hosted Machines
Hardware / disk
No
Cursor-hosted Cloud Agents
Managed infrastructure · elastic concurrency · all Cloud Agent features


1.
Do written policies require the repository checkout and tool execution to stay inside your perimeter?
This requirement usually comes from a compliance or security policy, not a team preference. The agent loop and inference remain in Cursor.
— Yes →
Self-Hosted Machines
Perimeter constraint
No
2.
Do agents need internal services that remain unreachable through Tailscale, PrivateLink,or egress allowlists?
Cursor-hosted agents can reach most private networks through Tailscale, PrivateLink, or egress allowlists.
— Yes →
Self-Hosted Machines
Network reach
No
3.
Do you need a custom OS, special hardware, or persistent local disk for a large repo?
Cursor-hosted agents run on Ubuntu VMs. Use a Dockerfile to customize tooling. Contact your enterprise account team for ARM support.
— Yes →
Self-Hosted Machines
Hardware / disk
No
Cursor-hosted Cloud Agents
Managed infrastructure · elastic concurrency · all Cloud Agent features


Agent loop • Cursor cloud
Inference (LLM)
Model inference
Prompt
Tool calls
Agent loop
Orchestration & state management
Start run
Stream to UI
Tool call
Results
Cursor UI
Web / desktop
Bridge
Frame routing
outbound HTTP/2
Tool execution • Your network
Outbound-only • No inbound from Cursor
agent-cli
Agent worker start
Prompt
Worker
Worker runtime
Prompt
Exec tool calls
Tools
Terminal
Filesystem
Browser
Forward
Return
Inference (LLM) loop


Agent loop • Cursor cloud
Inference (LLM)
Model inference
Prompt
Tool calls
Agent loop
Orchestration & state management
Start run
Stream to UI
Tool call
Results
Cursor UI
Web / desktop
Bridge
Frame routing
outbound HTTP/2
Tool execution • Your network
Outbound-only • No inbound from Cursor
agent-cli
Agent worker start
Prompt
Worker
Worker runtime
Prompt
Exec tool calls
Tools
Terminal
Filesystem
Browser
Forward
Return
Inference (LLM) loop


Start from scratch
Cloud
Plan, Build, / for ski
High Fast
Plan New Idea Tab
Multitask
Run on
Cloud
This Mac
Remote Machines
Search Remote Machines...
Self-Hosted Machines
lambda-test 0
cloud-demo 0
jacks-desktop 0
More
Connect via SSH



Register a local machine or VM as a self-hosted worker so Cursor cloud agents execute tool calls inside your network instead of on Cursor-hosted infrastructure.

```
$ cursor agent worker start
```

— Names CLI command, providers, and pool types with runnable example

`launch-20260902-a753d852` `snapshot-20260907`

---

02 Custom Modes pin skills as always-on chat modes NEW

80

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

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

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

— Exact keybinding and picker path with worked example `snapshot-20260907`

---

03 `/goal` command for long-lived agent objectives NEW

75

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

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

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

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

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

— Exact runnable command, though pairing mechanism is only briefly noted `snapshot-20260907`

---

04 **Origin: built-in code hosting service** NEW 75

Origin is a built-in code hosting service with repos, pull requests, code browsing, and GitHub sync, available in early beta on all paid plans. The Codebase tab is the home for Origin repos, with a `+New` button to create repos whose names become part of every repo URL at `cursor.com/codebase/<name>`; per-repo Settings manage GitHub sync status, access control, and connected apps. Cloud Agents can also start without a connected GitHub or third-party SCM provider via 'Start from scratch' in the repo picker, saving work to a Cursor Origin repo.

— Names URL pattern and UI paths but no API endpoints `snapshot-20260907`

---

05 **Projects: coordinator agent orchestrating subagents** NEW 70

Projects, accessible from the left-hand nav, runs a persistent coordinator agent that plans and delegates work across thousands of parallel subagents for large bodies of work spanning months, without requiring the user to stay online. Projects run cloud-by-default on persistent compute so work continues after closing your laptop, spinning up local agents on demand for machine-specific tasks; each subagent runs on its own isolated virtual machine with a clean copy of the project and independent cloud environment. A shared context file set syncs across all cloud and local agents in a Project, growing automatically as agents record codebase knowledge and working preferences.

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

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

— Describes mechanism and scope but no config or API names `launch-20260910-72494b2b`

snapshot-20260911 snapshot-20260907

06 **Subscriptions: Slack, schedule, and PR triggers for agents**

NEW 65

Subscriptions let a Project or cloud agent coordinator watch a Slack channel, run on a schedule, or follow all your PRs, acting autonomously on detected signals such as CI failures, opens, and merges without manual prompting.

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

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

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

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

– Names trigger types and a UI setup path snapshot-20260907 launch-20260910-72494b2b snapshot-20260911

07 **Mid-run follow-up steering for agents** NEW 65

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

– Names exact controls but no worked example shown snapshot-20260907 snapshot-20260911

08 **Builds: pre-prepared environment snapshots for cloud agents**

NEW 65

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

– Gives concrete speedup and dashboard location, no config key snapshot-20260907

09 **App extensions for Origin repos (Vercel, Depot, Buildkite)**

NEW 60

Origin repos gain a per-repo Apps tab for adding app extensions — Vercel, Depot, and Buildkite — enabling preview deployments and CI integration on every PR.

— Names integrations and effect but no config steps snapshot-20260907

THINNER COVERAGE BELOW

10 **Pull request sync between Origin and GitHub** NEW 45

Comments, reactions, and replies made on a pull request in either Origin or GitHub appear in the other within seconds.

— Clear behaviour described but no setup steps given snapshot-20260907

11 **Publish: live URL deployment for cloud agents** NEW 40

Publish connects a Vercel account to a cloud agent and produces a live URL from a built project.

— States mechanism briefly, no steps given snapshot-20260907

12 **Live port-forwarding for cloud agent environments** NEW 35

Adds in-browser live port-forwarding of a cloud agent's environment for real-time preview and design mode tooling.

— Thin description, no usage detail or config given snapshot-20260907

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

02 Pinecone

4 RELEASES · 2026-08-19 → 2026-09-11

Vector DBs & RAG

Pinecone is a vector database service that stores and searches high-dimensional embeddings for AI and machine learning applications.

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 4/5

Full-text search is GA with BM25 keyword ranking, Lucene query syntax, and 18-language stemming inside the same vector index — no separate search cluster — plus text-match filters that constrain semantic search to documents matching a literal keyword.

> Pinecone shipped full-text search (BM25) as a GA capability inside its vector index, alongside a broad expansion of administrative and access-control APIs covering organizations, projects, API keys, service accounts, and role bindings, plus a paginated document listing endpoint and a doubled pod-to-serverless migration limit.

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

01 Service account lifecycle and secret rotation APIs NEW

91

Adds full CRUD for service accounts: `POST /service-accounts` to create one with optional initial role bindings (client secret returned only once), `GET /service-accounts/{id}` and `GET /service-accounts` to retrieve or list them, `PATCH /service-accounts/{id}` to update a service account's name, and `DELETE /service-accounts/{id}` to delete one and revoke all tokens it minted within seconds. A rotate-secret endpoint rotates a service account's OAuth client secret, revoking the previous secret and its tokens within seconds and returning the new secret only once.

Rotate a compromised service account secret to immediately revoke old tokens and retrieve the new secret — returned only once.

```
$ curl -X POST 'https://api.pinecone.io/organizations/<org-id>/service-accounts/<service-account-id>/rotate-secret' \
  -H 'Api-Key: <api-key>'
```

Rotate a compromised service account OAuth secret immediately to revoke all outstanding tokens within seconds.

```
$ curl -X POST https://api.pinecone.io/service-accounts/<service-account-id>/rotate-secret \
  -H 'Authorization: Bearer <your-token>'
```

— Exact endpoints and methods with runnable rotate-secret commands. snapshot-20260911

snapshot-20260910

02 **Full-text search (BM25) inside vector index** NEW 88

Full-text search reaches general availability via the Documents API, running BM25 keyword ranking inside the same Pinecone Database index as dense and sparse vectors with no separate search cluster required. It supports Lucene query syntax (boolean operators, phrase queries, and fuzzy matching with the `~` operator) across multiple text fields per index, text-match filters that restrict semantic vector search results to documents matching a literal keyword filter, and tokenization/stemming in 18 languages plus language-agnostic n-gram tokenization for substring- and prefix-matching. Full-text search indexes support Dedicated Read Nodes for provisioned read capacity and Bring Your Own Cloud (BYOC) deployment, and are metered in the same read units and write units as vector indexes with no cluster sizing, sharding, or upgrade maintenance.

— Rich mechanism and named surfaces, but no runnable example given.

launch-20260909-87279580

03 **Document listing endpoint with pagination and prefix filter** NEW 81

Adds a List documents endpoint that returns up to 100 documents per page in a namespace, with an optional `prefix` parameter to filter by ID prefix and a `pagination` token for iterating through results.

Paginate through all documents in a namespace whose IDs share a common prefix — useful for auditing or bulk-processing a logical subset of records.

```
$ curl -X GET 'https://api.pinecone.io/namespaces/<namespace>/documents?prefix=user-&pagination=<token>' \
  -H 'Api-Key: <api-key>'
```

— Concrete page limit, parameters and a runnable example. snapshot-20260911

04 **Organization, project and API key administration APIs** NEW 68

Adds lifecycle management endpoints across three resource types: organizations (List organizations , Get organization details , Update an organization name, Delete an organization), projects (List projects , Create a new project , Get project details , Update a project including maximum Pod count and customer-managed encryption key (CMEK), Delete a project), and API keys (List API keys , Delete an API key).

– Endpoint names and fields given but no request examples. snapshot-20260911 snapshot-20260910

05 **Role-binding endpoint for access control** NEW 67

Adds `POST /role-bindings` to grant a role to a principal (such as a service account) at organization or project scope, letting a role be assigned immediately after creating the principal.

Grant a service account a scoped role at the project level immediately after creating it, using the new role-binding endpoint.

```
$ curl -X POST 'https://api.pinecone.io/organizations/<org-id>/role-bindings' \
  -H 'Api-Key: <api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"principal_id": "<service-account-id>", "role": "editor", "scope": "project", "project_id": "<project-id>"}'
```

– Named endpoint and scope options with a runnable example. snapshot-20260911 snapshot-20260910

THINNER COVERAGE BELOW

06 **Pod-to-serverless migration limit doubled** IMPROVED 45

Raises the pod-based index to serverless migration record limit to 50 million records, up from 25 million.

– Exact before/after numbers but no other mechanism given. snapshot-20260905

07 **Documentation index at /llms.txt** NEW 42

Serves a complete documentation index at `/llms.txt` for programmatic discovery of all available documentation pages.

– Single named endpoint with no further detail. snapshot-20260910

08 **List pending and expired invites** NEW 42

Adds `GET /invites` to list pending and expired invites in the caller's organization.

– Named endpoint only, minimal additional detail. snapshot-20260910

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

04 Braintrust

8 RELEASES · 2026-09-01

AI Observability & Evals

Braintrust is an AI evaluation platform that helps developers test, benchmark, and monitor AI applications and models in production.

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 4/5

Patterns runs scheduled Loop investigations that surface recurring failure modes and cost trends no scorer covers, saving each finding with supporting traces and a suggested fix, while the Debugger reports likely failure modes for a single trace citing the exact spans, tool calls, and model outputs behind each one.

> Braintrust's biggest change this window is a shift toward AI-driven trace investigation — Patterns runs Loop on a schedule to surface recurring problems, Debugger diagnoses single traces, and Loop itself moves to a persistent, server-side runtime that can edit prompts, scorers, and dashboards on its own. Alongside that, the SDKs picked up OpenAI Batch, ElevenLabs, and Agno instrumentation, the Go SDK gained prompt support, Java evals now run concurrently (with a breaking thread-safety requirement), and a new `glm-5.3-flash` model, blind human reviews, and dashboard sections shipped too.

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

01 Concurrent eval execution in Java SDK

BREAKING

95

Adds concurrent eval case execution to the Java SDK (v0.3.23): Eval.start() returns an `EvalResult` immediately while Eval.run() still awaits completion, with lifecycle APIs getStatus(), isDone(), awaitCompletion(), getStartedAt(), getDuration(), getCasesExecuted(), and getAbortCause(). Concurrency is controlled via .maxConcurrency(int) on `Eval.Builder` or `Devserver.Builder`, or a custom `Executor` via .executor(Executor) (default: 3 concurrent cases), with global defaults settable via `BRAINTRUST_DEFAULT_MAX_CONCURRENCY`, `BRAINTRUST_OTEL_MAX_QUEUE_SIZE`, `BRAINTRUST_OTEL_MAX_EXPORT_BATCH_SIZE`, and `BRAINTRUST_OTEL_EXPORT_INTERVAL_MILLIS`. As a result, `Task`, `Scorer`, and `Classifier` implementations must now be thread-safe.

Tune Java eval concurrency and OTel export behavior without touching code, useful in high-throughput CI environments.

```
$ export BRAINTRUST_DEFAULT_MAX_CONCURRENCY=10
export BRAINTRUST_OTEL_MAX_QUEUE_SIZE=4096
export BRAINTRUST_OTEL_MAX_EXPORT_BATCH_SIZE=512
export BRAINTRUST_OTEL_EXPORT_INTERVAL_MILLIS=2000
```

– Full API and env var names with breaking-change note and example `snapshot-20260911` `snapshot-20260910` `snapshot-20260909`

02 **Loop moves to a persistent managed runtime** IMPROVED

85

Loop now runs in a Braintrust-managed runtime (server-side, not browser), persisting threads and operating across logs, experiments, and datasets project-wide; it can create and edit prompts, scorers, datasets, facets, custom views, dashboards, and automations, with each write paused for approval unless auto-accept is enabled. It runs on built-in `GPT-5.6 Sol`, `GPT-5.6 Terra`, and `GPT-5.6 Luna` models (no AI provider config required), with GPT-6 Astra later added to the model picker including scheduled automations. An `org_name` parameter was added to Loop and the MCP server to disambiguate name lookups across organizations, resolving members, API keys, service tokens, patterns, prompts, and scorers by name instead of raw IDs.

– Names models and a config param but no runnable example `snapshot-20260911` `snapshot-20260910` `snapshot-20260909` `snapshot-20260908` `snapshot-20260907` `snapshot-20260906` `snapshot-20260904`

03 **OpenAI Batch API instrumentation in TypeScript SDK** NEW

85

Adds OpenAI Batch API instrumentation in the TypeScript SDK (v3.30.0) via `openaiFilesCreateTraced`, `openaiBatchesRetrieveTraced`, and `completeOpenAIBatchTrace`, tracing asynchronous OpenAI Batch requests end-to-end from file upload through submission and retrieval.

Trace an asynchronous OpenAI Batch request end-to-end so the full lifecycle — file upload, batch submission, and result retrieval — appears as linked spans in Braintrust.

typescript

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

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

Route an asynchronous OpenAI Batch request through Braintrust tracing so the full lifecycle appears in your experiment traces.

typescript

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

const file = await openaiFilesCreateTraced(openai, {
  file: fs.createReadStream('requests.jsonl'),
  purpose: 'batch',
});
const batch = await openaiBatchesRetrieveTraced(openai, file.id);
await completeOpenAIBatchTrace(batch);
```

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

typescript

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

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

Trace asynchronous OpenAI Batch requests end-to-end so each stage appears as a linked span in Braintrust.

typescript

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

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

Trace an asynchronous OpenAI Batch request end-to-end in Braintrust, from file upload through completion.

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

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

— Named functions with a full runnable code example `snapshot-20260911` `snapshot-20260909` `snapshot-20260908` `snapshot-20260907` `snapshot-20260906` `snapshot-20260904`

04 **Agno eval instrumentation in Python SDK** NEW 85

Adds Agno eval instrumentation to the Python SDK (v0.36.0) — `AccuracyEval`, `AgentAsJudgeEval`, `ReliabilityEval`, `PerformanceEval`, and eval suites are traced automatically via auto_instrument(); eval suite runs open a Braintrust experiment automatically.

Instrument Agno evals for automatic tracing and experiment creation in Braintrust, using the new auto_instrument() integration.

```python
import braintrust
from agno.eval.accuracy import AccuracyEval

braintrust.auto_instrument()

eval_run = AccuracyEval(agent=my_agent, dataset=my_dataset)
eval_run.run()
```

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

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

auto_instrument()

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

Instrument Agno eval suites so every eval run is automatically traced and opens a Braintrust experiment.

```python
from braintrust.integrations.agno import auto_instrument
from agno.evals import AccuracyEval, ReliabilityEval

auto_instrument()

AccuracyEval(agent=my_agent, inputs=test_inputs).run()
ReliabilityEval(agent=my_agent, inputs=test_inputs).run()
```

— Named eval classes and function with runnable example `snapshot-20260911` `snapshot-20260910` `snapshot-20260908` `snapshot-20260905`

---

05 SpanFilters gain error, metadata and duration filters NEW

85

Adds `has_error`, `metadata`, and `duration` fields to `SpanFilters` for trace.get_spans() in the Python SDK (v0.39.0), letting trace-level scorers filter spans by error state, metadata values, and execution time bounds.

Filter spans in a trace-level scorer to only those that errored and ran longer than 5 seconds, using the new `has_error` and `duration` SpanFilters fields.

```python
spans = trace.get_spans(SpanFilters(has_error=True, duration=
{"gt": 5.0}))
```

— Named fields and function with a runnable filter example `snapshot-20260911`

06 **PR comment filtering in GitHub eval action** NEW 80

Adds `report_scores` and `report_metrics` inputs to the GitHub eval action (v2.1.0) to filter which scores and metrics appear in the PR comment, each accepting a comma- or newline-separated list of names.

Filter a CI PR comment to show only specific scores and metrics — useful when your eval suite produces many signals but you only want pass-rate and latency surfaced in the pull request.

yaml

```yaml
- uses: braintrustdata/eval-action@v2.1.0
  with:
    api_key: ${{ secrets.BRAINTRUST_API_KEY }}
    report_scores: accuracy, pass_rate
    report_metrics: latency, cost
```

Restrict a PR comment to only the scores and metrics your team cares about, keeping the table focused during code review.

yaml

```yaml
- uses: braintrustdata/eval-action@v2
  with:
    report_scores: accuracy,faithfulness
    report_metrics: latency,cost
```

Limit a PR comment to only the scores and metrics that matter for your eval, avoiding noise from auxiliary checks.

yaml

```yaml
- uses: braintrustdata/eval-action@v2
  with:
    api_key: ${{ secrets.BRAINTRUST_API_KEY }}
    report_scores: accuracy, relevance
    report_metrics: latency, cost
```

Filter a PR comment to show only the scores and metrics you care about, reducing noise in high-signal CI pipelines.

```yaml
- uses: braintrust-dev/eval-action@v2.1.0
  with:
    api_key: ${{ secrets.BRAINTRUST_API_KEY }}
    report_scores: accuracy, faithfulness
    report_metrics: latency, cost
```

Filter a PR comment to only the scores and metrics you care about, keeping CI output concise for large eval suites.

```yaml
- uses: braintrustdata/eval-github-action@v2.1.0
  with:
    experiment_name: my-experiment
    report_scores: accuracy, coherence
    report_metrics: latency, cost
```

— Exact YAML config keys and workflow example provided `snapshot-20260908` `snapshot-20260907` `snapshot-20260905` `snapshot-20260904`

07 **New AI provider integrations in TypeScript SDK** NEW

80

Adds ElevenLabs instrumentation (`wrapElevenLabs`, TS SDK v3.32.0) for tracing text-to-speech and speech-to-text calls including streaming and timestamped audio, and `@google/generative-ai` instrumentation (`wrapGoogleGenerativeAI`, TS SDK v3.32.0) for the legacy Google Generative AI SDK (v0.24.x).

Instrument all ElevenLabs TTS/STT calls in a TypeScript agent for automatic tracing, including streaming and timestamped audio.

```typescript
import { wrapElevenLabs } from 'braintrust';
import ElevenLabs from 'elevenlabs';

const client = wrapElevenLabs(new ElevenLabs({ apiKey:
process.env.ELEVENLABS_API_KEY }));
```

— Named wrappers and versions with a runnable code example `snapshot-20260911`

08 **Patterns: scheduled AI trace investigation** NEW

75

Patterns runs Loop against a project on a schedule to surface recurring problems, cost trends, and cohort-level failure modes not covered by any scorer; each finding is saved

with supporting traces and a suggested fix, and can be used to create a scorer or classifier, continue investigation in Loop, or be copied as a prompt. Loop automations power Pattern discovery by default, running daily at 9:00 AM with timezone selection for custom cron schedules.

— Public-preview UI feature with clear mechanism, no exact command `snapshot-20260911`

`snapshot-20260910` `snapshot-20260909` `snapshot-20260908` `snapshot-20260907` `snapshot-20260905` `snapshot-20260904`

---

09 **GLM-5.3 Flash built-in model** NEW 75

Adds `glm-5.3-flash`, a multimodal reasoning model served by Braintrust with no AI provider setup required, selectable under the Braintrust provider in playgrounds, prompts, and scorers, or requestable via `glm-5.3-flash` through the Braintrust Gateway.

Use GLM-5.3 Flash via the Braintrust Gateway as a low-cost multimodal reasoning model without configuring a separate AI provider account.

```
$ curl https://api.braintrust.dev/v1/proxy/chat/completions \
  -H 'Authorization: Bearer <BRAINTRUST_API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "glm-5.3-flash", "messages": [{"role": "user",
"content": "Summarize this trace for me."}]}'
```

Use GLM-5.3 Flash as a zero-setup multimodal scorer in the Braintrust Gateway without configuring a third-party AI provider.

```
$ curl https://api.braintrust.dev/v1/gateway/chat/completions \
  -H 'Authorization: Bearer <BRAINTRUST_API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "glm-5.3-flash", "messages": [{"role": "user",
"content": "Score this response for accuracy: ..."}]}'
```

— Runnable Gateway curl example makes this immediately usable `snapshot-20260911`

`snapshot-20260910` `snapshot-20260909` `snapshot-20260904`

---

10 **Loop automations for scheduled recurring runs** NEW 70

Loop automations are scheduled Loop runs with their own instruction, model, and write permissions that produce a read-only thread you can Continue, with results deliverable to a Slack channel or webhook; automations support timezone selection for custom cron schedules.

— Mechanism is clear but setup is UI-based, no exact command `snapshot-20260911`

snapshot-20260910 snapshot-20260909 snapshot-20260908 snapshot-20260905 snapshot-20260904

11 **Harbor plugin verifier output upload** NEW 70

Adds standard verifier output upload to the Harbor plugin in the Python SDK (v0.36.0): `test-stdout.txt`, `test-stderr.txt`, and `ctrf.json`, with `attachments` and `redact_patterns` configuration.

– Named output files and config keys, but no example shown snapshot-20260911 snapshot-20260908 snapshot-20260905

12 **Scorer functions can omit name field** IMPROVED 65

Allows scorer functions in the Python SDK (v0.38.0) and TypeScript SDK (v3.31.0) to return `{"score": 0.5}` without a `name` field, using the function's name as the score key.

– Exact return syntax given, no separate runnable example snapshot-20260911 snapshot-20260910

13 **Prompt support in Go SDK** NEW 65

Adds prompt support to the Go SDK (v0.13.0): load a Braintrust prompt, render its variables locally, call any LLM client with the result, drive eval parameters, and link rendered calls back to their prompt in Braintrust traces.

– Workflow explained but no code example given snapshot-20260911

14 **Blind human reviews project setting** NEW 60

The 'Blind human reviews' project setting hides peer scores, comments, and aggregates until a reviewer submits their own scores; reviewers holding the project Update permission are always exempt.

– Named setting gives a clear starting point in project settings snapshot-20260911 snapshot-20260910 snapshot-20260907 snapshot-20260906 snapshot-20260905 snapshot-20260904

15 **Dashboard sections and shareable chart links** NEW 60

Adds named, reorderable, collapsible, duplicable dashboard sections with per-browser collapse-state memory (Pro and Enterprise plans), plus a shareable fullscreen link for individual dashboard charts, with chart move and copy actions now in each chart's more-options menu.

– UI feature list is specific but requires navigating the dashboard snapshot-20260911 snapshot-20260910 snapshot-20260909

THINNER COVERAGE BELOW

16 **Faster log search via Brainstore engine** IMPROVED 55

Adds faster log search via Brainstore's asynchronous execution engine, streaming results as segments finish for full-text, phrase, and multi-clause ANY_SPAN() (including negated) trace searches — no configuration required.

– Mechanism named but automatic, nothing for reader to configure `snapshot-20260911`

17 **Span metadata enrichment for Vercel AI SDK and Pydantic AI** IMPROVED 55

Adds `toolCallId` to Vercel AI SDK v6 tool call span metadata (TypeScript SDK v3.31.0), and adds invocation parameters (`temperature` , `max_tokens` , and similar model settings) to Pydantic AI LLM span metadata (Python SDK v0.37.0) when model settings are provided.

– Named fields across two integrations but no example shown `snapshot-20260911` `snapshot-20260910` `snapshot-20260907` `snapshot-20260904`

18 **Debugger for single-trace failure analysis** NEW 50

Debugger reports likely failure modes for a single trace, citing the spans, tool calls, and model outputs behind each one; 'Analyze trace' groups a run into labeled Work sections explaining what the agent did.

– Describes behavior but no exact navigation or command given `snapshot-20260911` `snapshot-20260910` `snapshot-20260907` `snapshot-20260904`

19 **Metric columnstore for high-volume projects** IMPROVED 50

Enable the Metric columnstore in project settings alongside log search optimization so Braintrust writes columnar artifacts during compaction, speeding up loading and sorting traces on the Logs page for high-volume projects.

– Clear setting to enable but no metrics on speedup `snapshot-20260911`

20 **input_audio content type in prompt templates** NEW 45

Adds `input_audio` content type support in Mustache and Nunjucks prompt template rendering in the TypeScript SDK (v3.32.0).

– Named content type but no usage example or mechanism detail `snapshot-20260911`

21 **Custom column name conflict rejection** IMPROVED 35

Custom column names that conflict with built-in table fields are now rejected at creation time.

– Bare behavior change with no scope or example `snapshot-20260908`

└──▷ **BREAKING ON UPGRADE**

! `Task`, `Scorer`, and `Classifier` implementations in the Java SDK (`v0.3.23`) must now be thread-safe, as eval cases run concurrently.

! `Task`, `Scorer`, and `Classifier` implementations in the Java SDK (v0.3.23) must now be thread-safe due to concurrent eval case execution.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

05 vLLM

2 RELEASES · 2026-09-07 → 2026-09-09

Local LLM Runtimes

A high-throughput and memory-efficient inference and serving engine for LLMs

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 4/5

vLLM v0.29.0 makes Model Runner V2 the default, adds admission-control queue flags, a sharded P2P weight-sync backend for RL, per-request spec-decode metrics, and five new model families, plus an out-of-tree Tenstorrent plugin that brings LLM serving to a new class of accelerator. The RL weight-sync and admission controls open serving patterns that previously needed custom infrastructure.

> vLLM's v0.29.0 release adds request admission-control flags, an RL weight-sync backend, deterministic decode tracing, five new model families, and removes ten legacy model architectures, while a separate Tenstorrent hardware plugin brings vLLM serving to TT-Metal accelerators with its own mesh scheduling and custom model registration.

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

01 Tenstorrent platform plugin with mesh scheduling controls NEW

95

Introduces the `vllm-tt-plugin` out-of-tree platform plugin that automatically registers Tenstorrent accelerators as a vLLM platform when `ttnn` from TT-Metal is importable, supporting `TTLlamaForCausalLM`, `TTMllamaForConditionalGeneration`, `TTQwen2ForCausalLM`, `TTQwen3ForCausalLM`, `TTQwen3_5ForConditionalGeneration`, `TTQwen2_5_VLForConditionalGeneration`, `TTQwen3VLForConditionalGeneration`, `TTMistralForCausalLM`, `TTMistral3ForConditionalGeneration`, `TTGemma3ForConditionalGeneration`, `TTGemma4ForCausalLM`, `TTGemma4ForConditionalGeneration`, `TTGemma4UnifiedForConditionalGeneration`, `TTDeepseekV3ForCausalLM`, and `TTGptOssForCausalLM`. Tenstorrent mesh targets are selected via the `MESH_DEVICE` setting (e.g. `TG`), which replaces `--tensor-parallel-size` / `-tp` / `-pp` (rejected outright on TT targets); `--additional-config.tt.sample_on_device_mode` (e.g. `all`) enables on-device sampling so the host receives already-chosen tokens rather than raw logits, and `--additional-config.tt.fabric_config` configures the mesh fabric topology (`FABRIC_1D`, `FABRIC_2D`, `FABRIC_1D_RING`). Scheduling uses a phase-constrained `TTScheduler` that issues exclusively prefill-only or decode-only steps (with chunked prefill and interleaved decode), and an alternative

`TTLaneCoordinator` scheduler class (`scheduler_config.scheduler_cls`) runs one independent `TTScheduler` per data-parallel lane for Galaxy-class single-execute models, with `TT_LLAMA_TEXT_VER` (e.g. `llama3_70b_galaxy`) and `TT_QWEN3_TEXT_VER` (e.g. `qwen3_32b_galaxy`) selecting Galaxy-optimised single-execute model variants.

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

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

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

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

— Fully named flags, env vars and schedulers with runnable examples

`launch-20260907-b130d8c8`

---

02 **Custom model registration for Tenstorrent plugin** NEW

85

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

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

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

— Named env vars with a runnable example showing exact usage `launch-20260907-b130d8c8`

---

03 **Request queue admission control flags** NEW

80

Adds `--max-num-queued-reqs` and `--max-num-queued-tokens` CLI flags for admission control on the request queue, capping queued load to prevent runaway queue growth during traffic spikes.

Cap queued load during traffic spikes to prevent runaway queue growth and enforce admission control.

```
$ vllm serve meta-llama/Llama-3.1-8B-Instruct --max-num-queued-reqs 500 --max-num-queued-tokens 131072
```

— Runnable example plus named flags, but mechanism is simple capping v0.29.0

04 **New model family support** NEW 75

Adds support for new models: Hy4-preview (Tencent 770B/49B-active MoE with Gated DeepSeek Sparse Attention and native MTP), Qwen3.8-Flash-Next (BF16/FP8/NVFP4 with MTP), GraniteSWA and GraniteMoeSWA, NemotronH_Omni_Reasoning_V3 with MTP, and Kimi K3 NVFP4 checkpoints.

— Names five model families with quant/format specifics, no example v0.29.0

05 **Sharded P2P weight sync backend for RL** NEW 70

Adds `sharded_rdt` P2P backend for RL weight sync, where each worker pulls only its TP/EP slice over NIXL or Ray Direct Transport.

— Clear mechanism described but no usage example v0.29.0

06 **Prefix cache and KV cache layout standardization** IMPROVED 70

Adds `prefix_cache_retention_interval` as a CLI argument (default `0`) to control Mamba prefix cache retention, makes prefix-cache `NONE_HASH` deterministic by default (removing the need to pin `PYTHONHASHSEED` for distributed KV cache users), and introduces a `KVCacheLayout` enum to standardize KV cache layout across backends.

— Three named surfaces with clear behavior change, no example v0.29.0

07 **FlashInfer backend integration expansion** NEW 65

Adds `--linear-backend flashinfer_cutedsl` opt-in flag to enable FlashInfer CuTeDSL BF16 low-latency GEMM, and enables FlashInfer all-reduce by default for tensor-parallel CUDA groups, with `VLLM_ALLREDUCE_USE_FLASHINFER=0` to opt out.

— Both flags named with default behavior, no runnable example v0.29.0

08 **CLI entrypoint and legacy backend deprecations** DEPRECATED 60

Removes the PyAV video decoder backend, deprecates `python -m vllm.entrypoints.openai.api_server` in favor of `vllm serve`, and removes the `VLLM_TEST_FORCE_FP8_MARLIN` environment variable.

— Names the deprecated entrypoint and removed env var precisely v0.29.0

THINNER COVERAGE BELOW

09 **LoRA support expansion across model families** NEW 55

Adds LoRA support for DeepSeek V4, Qwen3-Omni multimodal, and tower/connector LoRA for LLaVA-NeXT and LFM2-VL.

— Names four LoRA targets but no usage mechanism or example v0.29.0

10 **Model Runner V2 default rollout and new capabilities** IMPROVED 55

Model Runner V2 is now the default for all models, completing the rollout begun with pooling models; it also adds `extract_hidden_states` speculation support and prompt embeds support.

— Describes rollout status and two capabilities, no usage detail v0.29.0

11 **Multimodal input handling improvements** IMPROVED 55

Adds video embeds accepted by the Python frontend, modality-scoped `mm_processor_kwargs` honored for multimodal inputs, and ViT full CUDA graph support for Idefics3 and SmolVLM.

— Names three concrete additions but no mechanism depth or example v0.29.0

12 **Per-request speculative decode metrics** NEW 55

Adds `--per-request-spec-decode-metrics` CLI flag to expose per-request speculative decoding acceptance stats in OpenAI API responses.

— Named flag with clear purpose but no example v0.29.0

13 **Anthropic Messages API render endpoint** NEW 55

Adds `/v1/messages/render` endpoint for the Anthropic Messages API.

— Named endpoint but no method or usage detail given v0.29.0

14 **CPU offload extended to vision and audio towers** IMPROVED 50

Extends `--cpu-offload-params` support to vision and audio towers.

— Named flag scope extension, no further mechanism given v0.29.0

15 **Deterministic decode replay tracing** NEW 50

Adds `trace_decode_token_ids` for deterministic decode replay.

— Named config surface with minimal explanation of use v0.29.0

16 **FP8 quantization support for ModernBERT** NEW 45

Adds FP8 quantization support for ModernBERT.

– Names model and quant format, minimal further detail v0.29.0

17 **Legacy model architecture removals and migrations** BREAKING 45

Removes ten deprecated model architectures, and migrates FlexOlmo, Olmo3, and Hunyuan V1/VL to the Transformers modeling backend, removing their native implementations.

– Doesn't list the ten removed architectures by name v0.29.0

18 **Bidirectional attention for DeepSeek embedding models** NEW 35

Adds bidirectional attention for DeepSeek-backbone embedding models.

– Bare feature description without mechanism or example v0.29.0

19 **JIT warmup provider registry for attention backends** NEW 30

Adds a JIT warmup provider registry for attention backends.

– Bare description without usage detail or example v0.29.0

20 **Rank-local IPC weight updates for RL workflows** NEW 25

Adds rank-local IPC weight updates for RL workflows.

– Bare description with no mechanism or example v0.29.0

└──▷ **BREAKING ON UPGRADE**

! Ten deprecated model architectures are removed.

! FlexOlmo, Olmo3, and Hunyuan V1/VL are migrated to the Transformers modeling backend, removing their native implementations.

! The PyAV video decoder backend is removed.

! `python -m vllm.entrypoints.openai.api_server` is deprecated in favor of `vllm serve`.

! The `VLLM_TEST_FORCE_FP8_MARLIN` environment variable is removed.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

03 Anthropic

4 RELEASES · 2026-09-03 → 2026-09-11

Frontier Models

Anthropic develops Claude, an AI assistant API for developers to build applications with advanced language understanding and reasoning capabilities.

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 5/5

Claude Fable 5.1 launches with a 1M-token context window, 128k max output, and always-on adaptive thinking; ant apply brings declarative, file-based creation of agents, environments, and deployments with a lockfile for reproducible CI.

> Anthropic launched Claude Fable 5.1 and Claude Mythos 5.1 with 1M-token context windows and adaptive thinking, added declarative infrastructure-as-code via `ant apply`, and gave Claude Managed Agents an `auto` permission policy with live terminal session control, alongside several beta message-control headers, a full Admin API for the `ant` CLI and SDKs, and multiple breaking changes to tool_choice, thinking-block replay, and SDK beta clients.

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

01 Auto permission policy and live session control in Claude Managed Agents

NEW

95

Adds `auto` to Claude Managed Agents permission policies, letting the server evaluate each agent or MCP tool call and automatically run it, deny it, or pause for approval; `agent.tool_use` and `agent.mcp_tool_use` events gain `evaluation` and `evaluated_permission` fields reporting how each call was evaluated under the policy. The `ant` CLI adds `ant beta:sessions connect` to attach your terminal to a live session — following it, sending messages, and allowing or denying pending tool calls — plus a `--web` flag that serves the Claude Console session viewer locally and opens the session in a browser instead.

— Names policy, event fields, CLI command and flag precisely. `snapshot-20260911`

02 Admin API in ant CLI and SDKs NEW

95

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

— Names namespace, env vars, and covered resources exactly. `snapshot-20260907`

03 **Per-message effort control (beta)** NEW 90

Adds per-message effort control in beta on Claude Fable 5.1, Claude Mythos 5.1, and Claude Opus 5: include `output_config.effort` in a `role: "system"` message inside `messages` with the `mid-conversation-output-config-2026-07-01` beta header to change effort mid-conversation while preserving the prompt cache. Extended to Google Cloud for the same three models.

— Exact field, header and rollout scope given. `snapshot-20260907` `snapshot-20260910`

04 **Thinking block binding controls (beta)** NEW 85

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

— Names header and field but no usage example. `snapshot-20260907`

05 **Turn-scoped system messages (beta)** NEW 85

Adds turn-scoped system messages in beta via the `mid-conversation-system-clear-at-2026-08-21` header: setting `clear_at: "next_user_message"` on a mid-conversation `role: "system"` message makes it apply only to the current turn while remaining in history at no token cost.

— Exact header and field, no runnable example. `snapshot-20260907`

06 **Compliance API session coverage expands** IMPROVED 85

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

— Names exact product_surface values and required scope. `snapshot-20260907`

07 **SDK changes to beta files and skills clients** BREAKING 85

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

— Exact SDK versions and rename given, migration clear. snapshot-20260907

08 **Declarative `ant apply` for agent infrastructure** NEW

80

Adds the `ant apply` subcommand to the `ant` CLI (v1.30.0) for declarative, file-based creation and updating of agents, environments, skills, memory stores, and deployments from repository files; each run writes a `claude-lock.json` lockfile so subsequent runs — locally or in CI — update existing resources instead of creating duplicates.

— Names command and lockfile mechanism but no usage example. snapshot-20260907

September 3, 2026

09 **Claude Fable 5.1 and Claude Mythos 5.1 launch** NEW 75

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

— Full pricing and specs given but no request example. snapshot-20260907

10 **Streaming thinking updates via thinking.display** NEW

73

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

— Named field and header, no full example. snapshot-20260907

11 **Computer and browser use toolsets on Google Cloud** NEW

70

Makes the `computer_toolset_20260801` and `browser_toolset_20260801` computer use and browser use toolsets available on Google Cloud for Claude Fable 5, Claude Mythos 5, Claude Opus 5, Claude Sonnet 5, and Claude Opus 4.8.

— Named toolsets and models, no usage snippet. snapshot-20260907

12 **Thinking block replay restrictions** BREAKING 65

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

— States exact trigger and error, no workaround given. snapshot-20260907

13 **tool_choice restrictions on new models** BREAKING 60

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

— Exact error condition given, no migration path. snapshot-20260907

THINNER COVERAGE BELOW

14 **Automatic watermarking and content credentials** NEW 55

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

— Automatic behavior described; nothing for reader to configure. snapshot-20260907

15 **Personal and service account keys in Console** NEW 50

Adds personal keys and service account keys in the Console, scoped to a specific workspace or active across any workspace or admin endpoints the linked account can access.

— UI feature described briefly, no exact navigation. snapshot-20260907

16 **anthropic-version header now required** BREAKING 50

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

— Names affected APIs and header, no example value. snapshot-20260907

└──▷ **BREAKING ON UPGRADE**

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

! For accounts created on or after August 31, 2026, replaying a thinking block on Claude Fable 5.1 after the system prompt, tools, or an earlier message has changed returns a 400 error.

! Thinking blocks produced by Claude Fable 5.1 and Claude Mythos 5.1 are dropped by the API if replayed to an earlier model.

! In Python SDK 1.2.0, TypeScript SDK 0.122.0, Go SDK 1.68.0, Java SDK 2.59.0, Ruby SDK 1.67.0, and C# SDK 12.44.0, `client.beta.files` and `client.beta.skills` no longer send the `files-api-2025-04-14` and `skills-2025-10-02` beta headers and return the same response shapes as `client.files` and `client.skills`.

! In the SDK versions above, client.beta.skills.delete() now deletes a Skill together with all of its versions, and the beta Messages type `BetaSkill` (the container Skill reference) is renamed `BetaContainerSkill`.

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

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

// END OF TRANSMISSION

The Cyber Toolchain · This Week's Highlights · September 11, 2026
The full firehose — every feature release, every day — is in the newsletter.

▷ Subscribe