HEADS UP This site is currently under heavy development.

THE AI TOOLCHAIN NO. 006

Head

THE WEEK'S TOP PICKS

PUBLISHED SEPTEMBER 25, 2026 · FRIDAYS

// SHARE THIS WEEK ▽ Top 5 ▽ Top Pick

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

The best tooling updates that shipped this week.

▸ // HOW THESE PICKS ARE MADE SHOW

▾ // BENCHMARKS MOVED FOR TOOLS WE TRACK HIDE

Gemini API APEX-Agents Gemini 3.5 Flash

27.5% −4.0 pts released 2026-05-19 · leaderboard ↗

GLM APEX-Agents GLM-5.1

40.9% +5.3 pts released 2026-04-07 · leaderboard ↗

OpenAI APEX-Agents GPT-5.6 Terra max

58.2% +18.3 pts released 2026-07-09 · leaderboard ↗

Gemini API APEX-Agents Gemini 3.7 Flash

67.8% +36.3 pts released 2026-08-13 · leaderboard ↗

GLM APEX-Agents GLM-5.3

56.6% +21.0 pts released 2026-08-14 · leaderboard ↗

DeepSeek Models APEX-Agents DeepSeek V4 Pro 0813

47.3% +40.3 pts released 2026-08-13 · leaderboard ↗

MiniMax Models APEX-Agents MiniMax-M3

37.7% +31.5 pts released 2026-06-01 · leaderboard ↗

GLM APEX-Agents GLM-5.3-Flash

52.8% +17.2 pts released 2026-08-20 · leaderboard ↗

Qwen Models APEX-Agents Qwen3.5 397B-A17B

24.9% +11.3 pts released 2026-02-13 · leaderboard ↗

Gemini API APEX-Agents Gemini 3.6 Flash

46.9% +15.4 pts released 2026-07-21 · leaderboard ↗

Gemini API APEX-Agents Gemini 3.8 Flash

64.3% +32.8 pts released 2026-09-02 · leaderboard ↗

Anthropic OSWorld 2.0 Claude Opus 5 max

31.4% first result released 2026-07-24 · leaderboard ↗

Scores from Epoch AI Benchmarking Hub (CC BY 4.0), evaluated 2026-09-18 to 2026-09-25.

▾ // LOCAL LLM HARDWARE THIS WEEK HIDE

Minisforum MS-S1 Max list price changed **$3,799** +30% 2026-09-19 · source ↗

Vendor list prices, captured live and via the Wayback Machine; MSRP for cards until street capture ships. Events dated 2026-09-18 to 2026-09-25; each device page carries the full history.

▾ // 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: Pinecone can now run its data plane inside your cloud account without changing APIs; LangSmith turns traces into fine-tuning and review workflows; vLLM cuts serving startup friction and adds controls for larger fleets.

> GOVERN
>
> **Keep vector data in your own cloud account without rewriting retrieval code**
>
> Bring Your Own Cloud moves Pinecone's data plane into the customer's AWS, Google Cloud, or Azure account and region while preserving

Pinecone APIs and SDKs. Teams with residency, network, or procurement constraints can stop treating managed vector search as an exception architecture.

Pinecone

EVALUATE

Turn agent traces into review, red-team, and fine-tuning loops

LangSmith added `smithtune`, full-session Trajectories, and Engine v2 red teaming; Braintrust added automated trace investigation with Patterns, Debugger, Loop expansion, and Harbor eval sync. The practical shift is that failed or risky agent behavior can be inspected and recycled into evals or tuning data without first reconstructing the run by hand.

LangSmith · Braintrust

DEPLOY

Start and scale inference servers with less cold-start and fleet glue

vLLM added CUDA IPC weight caching, opt-in scale-out endpoints, serving flags, watermarking, KV offload, and Apple Silicon serving through vllm-metal. NVIDIA Dynamo added a Triton backend, experimental Sidecar, pluggable routing, tiered KV indexing, and TLS/mTLS, giving teams more of the machinery needed to run heterogeneous inference fleets without wrapping every engine themselves.

vLLM · NVIDIA Dynamo

GOVERN

Run coding agents in sandboxes with declared authority

Cloud Sandboxes let agent microVMs move between local and Docker-managed cloud compute, while the v3 OCI Kit spec declares what authority the agent has. That makes agent execution less dependent on ad hoc local setup and gives platform teams a concrete artifact to review for permissions, secrets, network policy, and MCP access.

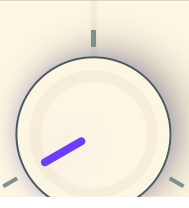
Docker Sandboxes

Does **GPT-5.5** read better? 👍 👎

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | **EVERYTHING**

Every change in this week's picks, in full. **all 91 changes**

Do you prefer **scroll**? 👍 👎

// THE PICKS

01	**vLLM**	Local LLM Runtimes
02	**Docker Sandboxes**	AI/LLM Security
03	**LangSmith**	AI Agent Frameworks
04	**Cursor**	AI Coding Agents
05	**NVIDIA Dynamo**	AI Model & Data Infrastructure

02 Docker Sandboxes

announced · Sep 24, 2026

AI/LLM Security

Docker Sandboxes isolates containerized applications in secure, lightweight environments for testing and development without affecting the host system.

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 5/5

Cloud Sandboxes lets teams launch and migrate microVM-isolated coding-agent environments onto Docker-managed cloud compute with the same CLI, centralized MCP access, secrets proxying, and network policy controls. That closes a real operations gap for running many agent sandboxes without provisioning machines or exposing raw credentials.

> Docker Sandboxes shipped Cloud Sandboxes for running and moving microVM - isolated agent sandboxes between local and Docker-managed cloud compute, plus a v3 OCI Kit specification for declaring agent authority. The window also tightened network-policy enforcement, expanded CLI controls for MCP , secrets, policies and JSON output, and introduced several breaking CLI and kit-compatibility changes.

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

01 Kit Specification v3 authority model NEW

97

Docker Sandbox Kit Specification v3, open source under Apache 2.0 at `docker/sandbox-kit-spec` , defines agent authority in ordinary OCI images via the `vnd.docker.sandbox.kit.descriptor` manifest annotation, covering network rules, credentials, volumes, ports, devices, and lifecycle hooks. It adds Kit kinds `workload` for the root filesystem and `mixin` for overlays supplying a CLI, network rule, credential binding, or context, composed through `provides` and `requires` declarations resolved by dependency graph rather than flag order. It also ships conformance suites for Kit artifacts and runtime implementation of normative capability-type pages, and `sbx run` launches Kits locally with `--kit` to compose mixins alongside a workload Kit.

Run a workload Kit and attach a GitHub CLI mixin from a local checkout to compose agent authority at launch time.

```
$ cd examples && sbx run ./hello --kit ./gh
```

— Spec annotation, kinds, CLI flag v0.45.0 launch-20260924-0c0143b0

02 Kit capability policy declarations NEW

97

V3 Kits support `com.docker.sandbox/network-policy@2` with `runtime.allow` and `runtime.deny` blocks specifying `hosts` , `methods` , and `paths` , with `deny` taking

precedence; they can scope network access by HTTP method and path, declare install-phase network access and credential use, and specify where an agent reads shared skills. They also support `com.docker.sandbox/credential@1` with `proxyManaged: true` and per-domain `inject` rules specifying `domain`, `header`, and `format`, so the runtime injects real credential values while only a sentinel exists inside the sandbox. `sbx kit validate` now rejects malformed API-key declarations, including invalid names, missing injection domains, invalid format placeholders, and Basic-auth usernames containing a colon, with warnings for no-effect or out-of-allowlist declarations.

Declare a mixin that limits a GitHub agent to read/write API access but explicitly blocks repository deletion, making the authority reviewable as a diff.

yaml

```yaml
capabilities:
  - type: com.docker.sandbox/network-policy@2
    config:
      runtime:
        allow:
          - github.com
          - hosts: [api.github.com]
            methods: [GET, HEAD, POST, PATCH, PUT, DELETE]
        deny:
          - hosts: [api.github.com]
            methods: [DELETE]
            paths: [/repos/**]
  - type: com.docker.sandbox/credential@1
    optional: true
    config:
      service: github
      phase: runtime
      apiKey:
        name: GH_TOKEN
        proxyManaged: true
      inject:
        - {domain: api.github.com, header: Authorization,
format: "Bearer %s"}
```

— Named capability types and fields `v0.45.0` `launch-20260924-0c0143b0`

03 **Cloud Sandboxes in sbx** NEW 95

Cloud Sandboxes run the same microVM isolation model as local sandboxes on Docker-managed infrastructure, with sessions up to 24 hours, parallel execution of many sandboxes, and no provisioning required. The CLI adds `sbx --cloud run <kit>` for launching cloud sandboxes, including `sbx --cloud run codex` and `sbx --cloud run claude`, plus `sbx move <project> --to cloud` to migrate a running local sandbox,

including its filesystem, to cloud compute and back. Cloud sandbox support requires `sbx` version 0.45.1 or later, and a web console can spin up, manage, and monitor Cloud Sandboxes without the CLI.

Start a Claude agent in the cloud before leaving for the day so it keeps running after you close your laptop.

```
$ sbx login
sbx --cloud run claude
```

Hand off a long-running task you started locally to cloud compute so it finishes overnight without tying up your machine.

```
$ sbx move my-project --to cloud
```

— Exact commands, limits, requirement, UI `launch-20260924-6c1aef17`

04 **Kit sets and authority gating** NEW 90

V3 adds `kind: set` descriptor support for naming a set of Kits, running full coherence checks at `docker buildx build` time, and merging them into one ordinary Kit, failing incoherent combinations at build instead of launch. Kit sets let authors pin a workload and mixins, add lifecycle hooks, capabilities, instructions, and arguments, and publish the result as a single OCI reference. Authority-diff gating records the normalized grant set on first launch and blocks upgrades that widen authority, including removing a `deny` rule, until a human approves; non-widening updates apply automatically.

— Concrete descriptor kind and build step `v0.45.0` `launch-20260924-0c0143b0`

05 **Network policy enforcement updates** IMPROVED 89

Network policy adds `--protocol` on local network allow rules so UDP can be selected explicitly alongside TCP, with experimental outbound UDP following sandbox network policy. Hostnames with a trailing dot are now treated as equivalent to their canonical form for routing, interception, and credential injection, and reverse-DNS lookups are restricted to destination IPs already authorized by policy, closing a previous policy bypass.

Allow outbound UDP to a specific host — for example, a custom DNS or VPN endpoint — while keeping TCP as the default for other rules.

```
$ sbx policy allow network udp://10.0.1.53:53 --protocol udp
```

— Exact flag and behavior changes `v0.45.0`

06 **Removed MCP catalog command** BREAKING 87

`sbx mcp catalog` has been removed. Register a remote MCP server with `sbx mcp add` and then authorize it with `sbx mcp auth`.

— Removed command and replacement `v0.45.0`

07 **Expanded CLI list and confirmation flags** IMPROVED 82

List commands for MCP server, secret, skill, template, volume, and policy-profile now support `--quiet` / `-q` for name-only output. Destructive commands add `--force`, and `sbx kit builder history rm` also adds `--yes` / `-y`, to skip confirmation prompts; declining a confirmation now returns a non-zero exit code. `sbx policy ls` adds `--created-via` and now shows how each rule was created, while `sbx settings` was added to `sbx --help` and the CLI reference.

— Flags named but commands grouped `v0.45.0`

08 **MCP transport registration controls** NEW 80

`sbx mcp add` adds `--disable-http2` to disable HTTP/2 for a remote MCP transport. This supports registering a remote MCP server when its HTTP/2 handling stalls long-lived streams.

Work around an MCP server whose HTTP/2 handling stalls long-lived streams by disabling HTTP/2 on registration.

```
$ sbx mcp add https://mcp.example.com/sse --disable-http2
```

— Exact command and flag `v0.45.0`

09 **Model provider configuration** NEW 77

The new `model.providers` setting lets `sbx run --model` target any OpenAI- or Anthropic-compatible endpoint. `sbx run opencode --model` also exposes a model's supported thinking levels as OpenCode variants, selectable with Ctrl+T.

Route `sbx run` to a self-hosted Anthropic-compatible endpoint instead of the default provider.

```
$ sbx run --model my-provider/claude-3-5-sonnet 'summarize this repo'
```

— Config key and command named `v0.45.0`

10 **Sandbox JSON output fields** IMPROVED 75

`sbx ls --json` now includes `created_at`, CPU limits, and memory limits. CPU and memory limits also appear in `sbx inspect --json`.

— Exact fields and JSON commands `v0.45.0`

11 **Cloud agent control surfaces** NEW 74

Cloud Sandboxes add Kits for pre-configured, pre-built sandbox images for Claude Code, Codex, Copilot, Antigravity, Open Code, and Hermes, plus user-defined kits. They also add a centralized MCP gateway so agents running in the cloud, locally, or in other clients such as the ChatGPT desktop app can reach configured MCP servers including Jira, Linear, Grafana, incident.io, or any streamable HTTP endpoint. A secrets proxy injects stored keys or tokens per request so agents never see raw secret values, and network policy controls define which endpoints each agent can reach, configured once and applied across all cloud sandboxes.

— Concrete integrations, limited operation detail `launch-20260924-6c1aef17`

12 **Stricter missing-resource errors** BREAKING 71

`sbx secret rm` now returns an error on stderr when the requested secret does not exist. `sbx mcp rm` now returns an error when the requested MCP server is not registered.

— Commands and error behavior named `v0.45.0`

13 **Secrets lifecycle improvements** IMPROVED 69

Running `sbx secret rm` without a service now opens an interactive picker showing existing local secrets with their scope, type, and name. Adding, updating, or removing global service secrets now propagates to existing local sandboxes immediately without a restart. Multiple OAuth-backed agents can now be composed in the same sandbox with credentials scoped to the kits that request them.

— Command named, behavior described `v0.45.0`

14 **V3 kit compatibility break** BREAKING 64

V3 workloads and mixins must be used together. They cannot be combined with v1 or v2 kits.

— Compatibility rule stated `v0.45.0`

15 **Cloud-local configuration separation** BREAKING 62

Local and cloud sandboxes keep separate secrets, templates, and network policies. Existing local workflows must be reviewed before being moved to cloud.

— Clear migration warning `launch-20260924-6c1aef17`

THINNER COVERAGE BELOW

16 **macOS Sbx.app distribution** IMPROVED 59

macOS distributions now ship as a single signed `Sbx.app` bundle. Homebrew and tarball PATH installs work through a symlink into the bundle.

— Packaging change, no migration steps `v0.45.0`

└──▷ BREAKING ON UPGRADE

! Local and cloud sandboxes keep separate secrets, templates, and network policies — existing local workflows must be reviewed before being moved to cloud.

! `sbx mcp catalog` has been removed; register a remote MCP server with `sbx mcp add` and then authorize it with `sbx mcp auth`.

! `sbx secret rm` now returns an error on stderr when the requested secret does not exist.

! `sbx mcp rm` now returns an error when the requested MCP server is not registered.

! V3 workloads and mixins must be used together; they cannot be combined with v1 or v2 kits.

announced · Sep 24, 2026

Docker Sandboxes gains Cloud Sandboxes: cloud-hosted microVM agents, `sbx --cloud run`, and `sbx move --to cloud` in one CLI.

└──▷ TRY IT

Start a Claude agent in the cloud before leaving for the day so it keeps running after you close your laptop.

```
$ sbx login
sbx --cloud run claude
```

Hand off a long-running task you started locally to cloud compute so it finishes overnight without tying up your machine.

```
$ sbx move my-project --to cloud
```

› Adds `sbx --cloud run <kit>` to launch a sandbox on Docker-managed cloud compute, e.g. `sbx --cloud run codex` or `sbx --cloud run claude`.

› Adds `sbx move <project> --to cloud` to migrate a running local sandbox — filesystem and all — to cloud compute without losing work (also works in reverse).

- Introduces Cloud Sandboxes: the same microVM isolation model as local sandboxes but running on Docker-managed infrastructure, supporting sessions up to 24 hours and parallel execution of many sandboxes with no provisioning required.
- Introduces Kits: pre-configured, pre-built sandbox images for leading coding agents (Claude Code, Codex, Copilot, Antigravity, Open Code, Hermes), plus support for user-defined kits.
- Adds a centralized MCP gateway so agents running in the cloud, locally, or in other clients (e.g. the ChatGPT desktop app) can all reach the same configured MCP servers (Jira, Linear, Grafana, incident.io, or any streamable HTTP endpoint).

+4 more

└──▷ BREAKING ON UPGRADE

! Local and cloud sandboxes keep separate secrets, templates, and network policies — existing local workflows must be reviewed before being moved to cloud.

announced · Sep 24, 2026

Docker Sandbox Kit Specification v3 ships: OCI images that declare agent authority via `vnd.docker.sandbox.kit.descriptor` annotations, enforced by the `sbx` CLI

└──▷ USE IT

Declare a mixin that limits a GitHub agent to read/write API access but explicitly blocks repository deletion, making the authority reviewable as a diff.

```yaml
capabilities:
  - type: com.docker.sandbox/network-policy@2
    config:
      runtime:
        allow:
          - github.com
          - hosts: [api.github.com]
            methods: [GET, HEAD, POST, PATCH, PUT, DELETE]
        deny:
          - hosts: [api.github.com]
            methods: [DELETE]
            paths: [/repos/**]
  - type: com.docker.sandbox/credential@1
    optional: true
    config:
      service: github
      phase: runtime
      apiKey:
        name: GH_TOKEN
        proxyManaged: true
      inject:
        - {domain: api.github.com, header: Authorization, format:
"Bearer %s"}
```

Run a workload Kit and attach a GitHub CLI mixin from a local checkout to compose agent authority at launch time.

```
$ cd examples && sbx run ./hello --kit ./gh
```

- Introduces the Docker Sandbox Kit Specification v3, open source under Apache 2.0 at `docker/sandbox-kit-spec`, defining a format for expressing agent authority (network rules, credentials, volumes, ports, devices, lifecycle hooks) inside an ordinary OCI image via the `vnd.docker.sandbox.kit.descriptor` manifest annotation.
- Adds two Kit kinds — `workload` (provides the root filesystem) and `mixin` (an overlay supplying a CLI, network rule, credential binding, or context) — composable via `provides` and `requires` declarations resolved by dependency graph, not flag order.
- Supports `com.docker.sandbox/network-policy@2` capability type with `runtime.allow` and `runtime.deny` blocks (specifying `hosts`, `methods`, and

`paths`) so a Kit can declare exactly which hosts and HTTP verbs an agent may reach, with `deny` taking precedence.

- Supports `com.docker.sandbox/credential@1` capability type with `proxyManaged: true` and per-domain `inject` rules (specifying `domain`, `header`, and `format`) so real credential values are injected by the runtime and only a sentinel exists inside the sandbox.
- Adds `kind: set` descriptor support: naming a set of Kits runs full coherence checks at `docker buildx build` time and merges them into one ordinary Kit, failing incoherent combinations at build rather than at launch.

+3 more

v0.45.0

Docker Sandboxes v0.45.0 ships v3 OCI kits with workload/mixin composition, new model provider config, and expanded CLI controls.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.45.0 https://github.com/docker/sbx-releases.git
# already have the repo? check out this version:
$ git checkout v0.45.0
```

└──▷ TRY IT

Route `sbx run` to a self-hosted Anthropic-compatible endpoint instead of the default provider.

```
$ sbx run --model my-provider/claude-3-5-sonnet 'summarize this repo'
```

Allow outbound UDP to a specific host — for example, a custom DNS or VPN endpoint — while keeping TCP as the default for other rules.

```
$ sbx policy allow network udp://10.0.1.53:53 --protocol udp
```

Work around an MCP server whose HTTP/2 handling stalls long-lived streams by disabling HTTP/2 on registration.

```
$ sbx mcp add https://mcp.example.com/sse --disable-http2
```

› Adds `model.providers` setting so `sbx run --model` can target any OpenAI- or Anthropic-compatible endpoint.

› Adds `sbx kit validate` rejection of malformed API-key declarations, including invalid names, missing injection domains, invalid format placeholders, and Basic-auth usernames containing a colon, with warnings for no-effect or out-of-allowlist declarations.

› Adds `--quiet` / `-q` flag to MCP server, secret, skill, template, volume, and policy-profile list commands for name-only output.

› Adds `--protocol` flag to local network allow rules so UDP can be selected explicitly alongside TCP (experimental outbound UDP now follows sandbox network policy).

› Adds `--created-via` filter flag to `sbx policy ls`, which now also shows how each rule was created.

+14 more

└──▷ BREAKING ON UPGRADE

! `sbx mcp catalog` has been removed; register a remote MCP server with `sbx mcp add` and then authorize it with `sbx mcp auth`.

! `sbx secret rm` now returns an error on stderr when the requested secret does not exist.

! `sbx mcp rm` now returns an error when the requested MCP server is not registered.

! V3 workloads and mixins must be used together; they cannot be combined with v1 or v2 kits.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

03 LangSmith

announced · Sep 24, 2026

AI Agent Frameworks

LangSmith is a platform for debugging, testing, and monitoring LangChain applications and language model workflows in production.

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 5/5

LangSmith can now turn production agent trajectories into supervised fine-tuning datasets, train open models through Baseten or Fireworks, evaluate them in LangSmith, and deploy the tuned model with the new `smithtune` CLI. The same release also adds trajectory-level review/eval views, Engine v2 red teaming from production traces and repos, Managed Deep Agents user memory, built-in web search, and Custom Apps.

> LangSmith's biggest additions were an end-to-end fine-tuning workflow built around `smithtune`, Trajectories for reviewing full agent sessions, and Engine v2 red teaming. The window also added concrete operator and API surfaces including `GET /v2/threads/stats`, `LICENSE_EXPIRATION_WARNING_DAYS`, `ls_provider`, annotation-queue item APIs, Jev judge support, and several LLM Gateway, OAuth, dataset, Playground, and Managed Deep Agents updates.

└──▷ WHAT SHIPPED · 30 FEATURES most completely described first ? what's the number?

01 Fine-tuning CLI pipeline NEW

97

LangSmith added Fine-Tuning for supervised fine-tuning of open models on task-specific examples from trajectory data, centered on the `smithtune` CLI. The workflow includes `smithtune plan` to review model, number of examples, learning rate, batch size, and epochs; `smithtune train` to submit a LoRA SFT job to Fireworks managed SFT or Baseten Loops with automatic checkpoint selection by lowest validation loss; `smithtune evaluate` to replay-evaluate the base model against the fine-tuned checkpoint and upload results to LangSmith as a comparison link; and `smithtune deploy` to serve the tuned model and connect it to an application. It also uploads curated golden trajectories to a persistent LangSmith dataset artifact for auditability and training/evaluation splits, and supports Fireworks and Baseten backends with only an API key.

– CLI commands, backends, and workflow are named `launch-20260925-ca933f1f` `launch-20260924-2581abc3`

02 Annotation queue item API NEW

96

LangSmith added `POST /annotation-queues/<id>/items` to add items to an annotation queue. The endpoint supports `item_type: RUN` and `item_type: THREAD`, where thread items require `thread_id` and `session_id`; mixed RUN and THREAD

batches are supported, the server resolves runs via ClickHouse or SmithDB, responses use a standards-shaped items envelope, and requests are limited to 200 items with a validation error when exceeded.

Add a batch of conversation threads to an annotation queue for human review, mixing thread and run items in one request.

```
$ curl -X POST 'https://api.smith.langchain.com/annotation-
queues/<queue_id>/items' \
  -H 'Content-Type: application/json' \
  -H 'x-api-key: <LANGSMITH_API_KEY>' \
  -d '{"items": [{"item_type": "THREAD", "thread_id": "
<thread_id>", "session_id": "<session_id>"}, {"item_type":
"RUN", "run_id": "<run_id>"}]}'
```

— Endpoint, fields, limit, and example provided `snapshot-20260918`

03 **Thread statistics API** NEW 94

LangSmith added `GET /v2/threads/stats` to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project. It is available on self-hosted ClickHouse-backed deployments, including when SmithDB queries are disabled, and thread query clients can request server-sent events on `/v2/threads/stats` to receive results progressively while preserving cursor pagination.

Pull aggregated latency, token, and cost statistics for a tracing project to feed into a cost-monitoring dashboard.

```
$ curl -X GET 'https://<langsmith-host>/v2/threads/stats?
project_id=<project_id>' \
  -H 'Authorization: Bearer <api_key>'
```

Pull thread statistics including latency, token usage, and cost for a tracing project to feed into a security posture dashboard or cost audit.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

Pull aggregated latency, token, and cost statistics for a tracing project to feed a security-operations dashboard or cost-alerting pipeline.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

Pull thread-level latency, token usage, and cost statistics for a tracing project in a single API call.


```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```


– Exact endpoint and curl examples included `snapshot-20260925` `snapshot-20260924` `snapshot-20260922` `snapshot-20260920` `snapshot-20260919`

---

04 **Gateway provider metadata** IMPROVED 90

LLM Gateway OpenAI-compatible model configurations now accept `ls_provider` to populate trace metadata and match custom models to provider-specific pricing. Examples show `ls_provider` values such as `openai` and `bedrock`, with model configuration fields including `model`, `base_url`, `ls_provider`, and in one example `api_key` using `${LANGSMITH_GATEWAY_API_KEY}`.

Route a custom OpenAI-compatible model through the LLM Gateway with correct provider metadata and pricing attribution in traces.

```json
{
  "model": "my-custom-model",
  "base_url": "https://my-endpoint.example.com/v1",
  "ls_provider": "openai"
}
```

Tag a custom Bedrock model in the LLM Gateway so its traces carry provider metadata and map to provider-specific pricing.

```json
{
  "model": "my-bedrock-llama3",
  "base_url": "https://bedrock.us-east-1.amazonaws.com",
  "ls_provider": "bedrock"
}
```

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"
}
```

Tag a custom OpenAI-compatible model with its provider so the LLM Gateway populates trace metadata and applies the right pricing.

```json
{
  "model": "my-custom-gpt4o",
  "base_url": "https://my-endpoint.example.com/v1",
  "ls_provider": "openai"
}
```

Tag a custom OpenAI-compatible model with its provider so the LLM Gateway records accurate trace metadata and applies provider-specific pricing.

```json
{
  "model": "my-custom-model",
  "ls_provider": "openai",
  "base_url": "https://my-gateway.example.com/v1"
}
```

Tag an OpenAI-compatible custom model with `ls_provider` so Gateway traces carry the correct provider label and map to provider-specific pricing.

```json
{
  "model": "my-custom-gpt4",
  "base_url": "https://my-gateway.example.com/v1",
  "ls_provider": "openai"
}
```

Route a custom OpenAI-compatible model through the LLM Gateway and tie its traces to provider-specific pricing by setting `ls_provider`.

```json
{
  "model": "my-custom-model",
  "base_url": "https://gateway.langsmith.com/oai/v1",
  "ls_provider": "openai",
  "api_key": "${LANGSMITH_GATEWAY_API_KEY}"
}
```

— Config field and JSON examples are present `snapshot-20260925` `snapshot-20260922` `snapshot-20260919`

05 **LLM-as-judge evaluator CLI** NEW 87

LangSmith added `langsmith evaluator create-llm` to define structured LLM-as-judge evaluator rules from a prompt, schema, and model config file, targeting a project or dataset. It also added `evaluator upload --replace`, which updates existing code evaluator rules in place and avoids a delete-before-create window if the replacement upload fails.

— CLI commands and behavior are explicit `snapshot-20260918`

06 **Jev judge model integration** NEW 87

LangSmith added TypeSafe's Jev as a judge model for online evaluations. Configure TypeSafe AI under Settings > Provider secrets with `TYPESAFE_API_KEY`, then select `jev-latest` under the TypeSafe provider in LLM-as-a-Judge evaluator configuration; Jev supports noul, choice, and score question types, and typed questions map to feedback keys in the Evaluators tab of any tracing project for filtering, charting, alerts, and automations.

— Secret, model name, UI locations named `launch-20260921-c4267686`

07 **Runs rule thread testing** IMPROVED 85

The `/runs/rules/validate` endpoint now supports thread evaluators by accepting `test_thread_id` and `session_id` to test a multi-turn evaluator against a real conversation before saving. Multi-turn evaluators also gained a Test action that runs the evaluator against a sample thread before saving the rule.

— Endpoint and request fields are named `snapshot-20260918`

08 **Self-hosted license warning config** NEW 84

Self-hosted LangSmith operators can set `LICENSE_EXPIRATION_WARNING_DAYS` to configure how many days before license expiration the warning banner appears. The default is 7 days, and examples set `LICENSE_EXPIRATION_WARNING_DAYS=14` or `LICENSE_EXPIRATION_WARNING_DAYS=30` in `.env`.

Give self-hosted operators earlier notice of license expiry — e.g. 30 days — so the team has time to renew before the banner becomes urgent.

```env
LICENSE_EXPIRATION_WARNING_DAYS=30
```

Set a 14-day lead time on the license expiration banner so self-hosted teams have more runway to renew before the warning appears.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

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
```

Warn the ops team two weeks before a self-hosted LangSmith license expires instead of the default seven days.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Give self-hosted operators earlier notice of an approaching license expiration to avoid unexpected access interruptions.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Set the license-expiration warning banner to appear 14 days early on a self-hosted deployment so ops teams have more runway before the instance goes dark.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Set the license-expiration warning horizon to 14 days on a self-hosted deployment so the ops team gets earlier notice.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

— Environment variable, default, and values named `snapshot-20260925` `snapshot-20260922` `snapshot-20260919`

09 **Dataset export permission split** BREAKING 84

Dataset CSV and JSONL exports now require both dataset read and download permissions, and direct API requests without download permission return a forbidden response. The read-datasets permission was split into a new `download datasets` permission, and the download button is disabled for users without it, so existing API clients or tokens with only dataset read permission can no longer export datasets.

— Permission names and migration impact included `snapshot-20260925` `snapshot-20260924` `snapshot-20260923` `snapshot-20260922` `snapshot-20260921` `snapshot-20260920` `snapshot-20260919` `snapshot-20260918`

10 **Trajectories session review** NEW 81

LangSmith introduced Trajectories, a chronological conversational view of an agent session that aggregates messages from humans, AI, and tools across the main agent and subagents. It supports traces from LangChain, LangGraph, Deep Agents, OpenAI and Claude SDKs, and coding agents including Codex, Claude Code, and Cursor; trajectories can be routed to annotation queues, scored by online evaluators, flagged and annotated by subject-matter reviewers, saved to datasets, and exported for supervised fine-tuning workflows.

— Scope and supported sources are named `launch-20260925-ca933f1f` `launch-20260924-aff0eb63`

11 **Feedback formulas deprecation** DEPRECATED 80

The legacy feedback formula endpoints `POST/GET /feedback/formulas` and `GET/PUT/DELETE /feedback/formulas/{feedback_formula_id}` are deprecated and scheduled for removal on 2026-08-20. Existing feedback formulas should be migrated to composite evaluators before that date.

— Endpoints and removal date are explicit `snapshot-20260918`

12 **Gateway prompt portability** IMPROVED 79

Prompts saved with a LangSmith Gateway model now use a standard `ChatOpenAI` manifest with the unified Gateway URL and a `LANGSMITH_GATEWAY_API_KEY` secret reference. The saved prompts are compatible with standard LangChain SDKs.

— Manifest and secret reference named `snapshot-20260925` `snapshot-20260922`

snapshot-20260919

13 **Dataset comparison SDK removal** BREAKING 78

Legacy dataset comparison helpers were removed from the public OpenAPI spec and generated SDKs. Use `POST /v2/datasets/<id>/experiment-runs` for paginated experiment comparison going forward.

— Replacement endpoint is named snapshot-20260918

14 **Custom Apps workspace interfaces** NEW 72

LangSmith introduced Custom Apps so teams can build, publish, run, and share custom interfaces on top of LangSmith data such as traces, experiments, and annotations inside the LangSmith workspace. Apps can be created with templates, in-app chat, LangSmith Chat, or code using the LangSmith API with coding-agent guidance; Plus plans include one Custom App per organization and Enterprise plans include unlimited Custom Apps.

— Builder options and plan limits included launch-20260925-ca933f1f

launch-20260924-2b947758

15 **OAuth credential ownership** IMPROVED 72

OAuth authorization sessions now accept `owner_type` and `owner_id`, allowing a human to authorize a managed credential for either a user or an agent. They also accept workspace owners, enabling one managed OAuth credential to be shared across a workspace, and agent OAuth connections now show a provider-branded result page after authorization to distinguish successful from failed connection attempts.

— Parameters and ownership scopes named snapshot-20260925 snapshot-20260922

snapshot-20260919

16 **Managed Deep Agents v0.8** IMPROVED 71

Managed Deep Agents v0.8 added user-level memory scoped to each authenticated user, separate from shared agent-level memory, with per-level access policies. It also added agent-owned and user-owned credentials for external connections such as GitHub and Notion, Slack channel file transfer for logs, spreadsheets, contracts, and screenshots, an HTTP channel for services that can send a JSON webhook, and built-in web search as a prebuilt tool powered by Parallel without a separate vendor account, API key, or custom tool setup.

— Capabilities named but few exact setup surfaces launch-20260925-ca933f1f

17 **Playground model controls** IMPROVED 68

Amazon Bedrock models in the Playground now expose request timeout and max retries options, removing the previous hard cutoff at the AWS client default of 60 seconds for

long-running agent calls. Claude Fable 5.1 is now available in the Anthropic, Bedrock, and Vertex AI model selectors in the Playground.

— UI area, providers, and 60-second cutoff named `snapshot-20260925` `snapshot-20260922` `snapshot-20260919`

18 **Thread dataset actions** IMPROVED 66

LangSmith added Add to dataset and Add to annotation queue buttons to the thread header in every thread view: Messages, Turns, and Details. The thread `add-to-dataset` popup also gained a New Dataset option that creates an empty dataset and adds the selected threads to it.

— Buttons and thread views are named `snapshot-20260925` `snapshot-20260922` `snapshot-20260919`

19 **Experiment split column** IMPROVED 65

The experiment comparison view gained an optional, reorderable Splits (latest) column. It shows each example's current dataset split assignments as chips and reflects live membership rather than the as-of-run snapshot.

— Column name and behavior are clear `snapshot-20260918`

20 **Engine v2 red teaming** NEW 62

LangSmith Engine v2 red teaming uses production traces and repos to generate hypotheses about unreported issues, test them, and surface confirmed failures for review before they reach production. Engine v2 also expands issue detection beyond errors and unmet user requests to performance trends such as error rate, latency, and cost, plus inefficient agent patterns such as repetitive tool calls or unnecessarily long trajectories; the next self-hosted LangSmith release will support BYOK for Engine.

— Mechanism stated but no user surface `launch-20260925-ca933f1f`

21 **LLM Gateway fallback timeouts** IMPROVED 61

LLM Gateway fallback chains can now advance to the next model when a candidate does not return response headers within a configured timeout. Active response streams continue after headers arrive, and Gateway requests authenticated with a session or OAuth bearer token now retain authentication and workspace context across internal routing, including model fallbacks.

— Behavior described, timeout key not named `snapshot-20260925` `snapshot-20260923` `snapshot-20260922` `snapshot-20260919`

22 **Code evaluator type support** IMPROVED 60

Code evaluator upload now accepts Python entrypoints annotated with PEP 604 union return types, for example `-> dict | None`.

– Exact accepted annotation example included snapshot-20260918

THINNER COVERAGE BELOW

23 **Private ECR IAM roles** IMPROVED 56

Commercial private ECR registries in Sandboxes can now assume an AWS IAM role on demand. This enables repository and tag search without expiring saved authorization tokens.

– Integration and behavior named, setup not provided snapshot-20260925 snapshot-20260922 snapshot-20260919

24 **Trajectory evaluation isolation** IMPROVED 55

Trajectory batch reads now run on a dedicated in-cluster gRPC service instead of the smith-go monolith. This isolates load from trajectory evaluations so it no longer competes with UI and API traffic.

– Before/after architecture is described snapshot-20260925 snapshot-20260922 snapshot-20260919

25 **MCP connector client metadata** IMPROVED 52

LangSmith MCP connectors now use Client ID Metadata Documents when supported. Dynamic client registration remains as a compatibility fallback.

– Mechanism named, no configuration surface snapshot-20260925 snapshot-20260922 snapshot-20260919

26 **Workspace tag assignment** IMPROVED 49

Workspace editors can now assign resources to an Application tag value from Settings > Resource Tags.

– UI path is included snapshot-20260923 snapshot-20260920

27 **Managed agent connection errors** IMPROVED 47

Managed Deep Agent channel replies now name a missing workspace connection and show how an agent developer can configure it, instead of reporting a generic run failure.

– Behavioral improvement, no exact UI or config snapshot-20260922

28 **IssueBench Opus 5 benchmark** NEW 46

IssueBench now includes a benchmark task that compares pre- and post-Opus 5 agent traces for cost and trajectory regressions.

– Task is named, usage absent snapshot-20260925 snapshot-20260922 snapshot-20260919

29 **Workspace ID switching** IMPROVED 42

The workspace switcher now accepts workspace IDs in addition to workspace names.

— Small UI input change only `snapshot-20260925` `snapshot-20260922` `snapshot-20260919`

---

30 **Project trace limits** NEW 38

LangSmith now enforces user-defined monthly trace limits scoped to individual projects and users.

— Scope named, configuration not described `snapshot-20260918`

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read AND download permissions — existing API clients that have dataset read but not download permission will receive a forbidden response instead of the export.

! Dataset CSV and JSONL exports now enforce a download permission check; API callers that previously exported datasets with only read permission will receive a forbidden response.

! 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.

! Dataset CSV and JSONL exports now require both dataset read and download permissions; existing API clients that have read-only permission will receive a forbidden response on export requests.

! Dataset CSV and JSONL exports now require both dataset read and download permissions; API callers that previously exported with read-only tokens will receive a forbidden response.

! Dataset CSV and JSONL exports now require both dataset read AND download permissions — users or API tokens that previously exported with only dataset read permission will receive a forbidden response.

! Dataset CSV and JSONL exports now require both dataset read and download permissions; existing API integrations that rely on dataset read permission alone will receive a forbidden response.

! The legacy feedback formula endpoints (`POST/GET /feedback/formulas` and `GET/PUT/DELETE /feedback/formulas/{feedback_formula_id}`) are deprecated and scheduled for removal on 2026-08-20; migrate existing feedback formulas to composite evaluators before that date.

! Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; use `POST /v2/datasets/<id>/experiment-runs` for paginated experiment comparison going forward.

`announced · Sep 24, 2026`

LangSmith launches Fine-Tuning with the `smithtune` CLI to turn agent trajectories into custom fine-tuned models via Fireworks or Baseten.

› Adds `smithtune plan` to review training settings (model, number of examples, learning rate, batch size, epochs) before committing to a fine-tuning run.

- Adds `smithtune train` to submit a LoRA SFT job to Fireworks managed SFT or Baseten Loops, with automatic checkpoint selection based on lowest validation loss.
- Adds `smithtune evaluate` to run a built-in replay evaluation comparing the base model against the fine-tuned checkpoint, with results uploaded to LangSmith as a comparison link.
- Adds `smithtune deploy` to serve the tuned model and connect it to an application after evaluation.
- Integrates with Fireworks and Baseten as training backends, requiring only an API key for either platform — no GPU provisioning or training infrastructure to manage.

+1 more

`announced · Sep 24, 2026`

LangSmith adds Trajectories: a chronological, conversational view of agent sessions for debugging, review, and online eval scoring.

- Supports Trajectories out of the box for traces from LangChain, LangGraph, Deep Agents, OpenAI and Claude SDKs, and coding agents including Codex, Claude Code, and Cursor.
- Enables routing trajectories to annotation queues so subject-matter experts can score agent behavior and flag issues in a readable session view, without parsing nested traces.
- Allows online evaluators to score trajectories, giving evaluators a deduplicated, ordered session view instead of accumulating repeated turn-over-turn context.
- Supports saving high-quality trajectories to datasets and exporting them for supervised fine-tuning workflows, using production sessions as post-training examples.

`announced · Sep 25, 2026`

LangSmith adds Engine v2 red teaming, Managed Deep Agents v0.8 user-memory and web search, Trajectories, Fine-Tuning CLI, and Custom Apps.

- Adds `smithtune` CLI to turn LangSmith trajectories into fine-tuning datasets, train open models via Baseten or Fireworks, evaluate results with LangSmith, and serve tuned models — handling the full supervised fine-tuning (SFT) pipeline without a custom build.
- Introduces LangSmith Engine v2 red teaming: uses production traces and repos to generate hypotheses about unreported issues, tests them, and surfaces confirmed failures for review before they reach production.
- Managed Deep Agents v0.8 adds user-level memory so agents store context scoped to each authenticated user, separate from shared agent-level memory, with per-level

access policies.

- Managed Deep Agents v0.8 adds support for both agent-owned and user-owned credentials for external connections such as GitHub and Notion.
- Managed Deep Agents v0.8 expands Slack channel support to include file transfer (logs, spreadsheets, contracts, screenshots).

+8 more

announced · Sep 24, 2026

LangSmith: Adds Custom Apps, letting teams build and publish custom UI interfaces on top of LangSmith data (traces, experiments, annotations) directly inside the LangSmith workspace, eliminating the need to self-host auth, permissions, or sharing.

- Adds a chat-driven app builder via LangSmith Chat (Plus and Enterprise plans) where users describe the data and visualization they want and Chat generates a working Custom App.
- Supports building Custom Apps in code using provided templates and the LangSmith API, with coding-agent guidance.
- Plus plans include one Custom App per organization; Enterprise plans include unlimited Custom Apps, both with the LangSmith Chat experience.

announced · Sep 21, 2026

LangSmith adds TypeSafe's Jev as a judge model for online evaluations, enabling fast, low-cost typed scoring of agent traces.

- Adds `TYPESAFE_API_KEY` provider secret (under Settings > Provider secrets) to connect TypeSafe AI as a model provider in LangSmith.
- Adds `jev-latest` as a selectable model under the TypeSafe provider in LLM-as-a-Judge evaluator configuration, supporting noul (yes/no probability), choice, and score question types.
- Supports Jev-as-a-judge evaluators in the Evaluators tab of any tracing project, where each typed question maps to its own feedback key that can be filtered, charted, or used to trigger alerts and automations.

snapshot-20260925

LangSmith: Adds `GET /v2/threads/stats` endpoint to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project, including on self-hosted ClickHouse-backed deployments.

└──▷ **USE IT**

Give self-hosted operators earlier notice of license expiry — e.g. 30 days — so the team has time to renew before the banner becomes urgent.

```env
LICENSE_EXPIRATION_WARNING_DAYS=30
```

Route a custom OpenAI-compatible model through the LLM Gateway with correct provider metadata and pricing attribution in traces.

```json
{
  "model": "my-custom-model",
  "base_url": "https://my-endpoint.example.com/v1",
  "ls_provider": "openai"
}
```

- Adds `LICENSE_EXPIRATION_WARNING_DAYS` environment variable for self-hosted operators to configure how many days before license expiration the warning banner appears (default: 7).
- Sets `ls_provider` on OpenAI-compatible model configurations to populate trace metadata and match custom models to provider-specific pricing through the LLM Gateway.
- LLM Gateway fallback chains can now advance to the next model when a candidate does not return response headers within a configured timeout, while active response streams continue after headers arrive.
- OAuth authorization sessions now accept `owner_type` and `owner_id` parameters, allowing a human to authorize a managed credential for either a user or an agent.
- OAuth authorization sessions now accept workspace owners, allowing one managed OAuth credential to be shared across a workspace.

+13 more

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read AND download permissions — existing API clients that have dataset read but not download permission will receive a forbidden response instead of the export.

`snapshot-20260924`

LangSmith adds `/v2` /threads/stats endpoint, IAM role-based ECR access, `LICENSE_EXPIRATION_WARNING_DAYS` config, and expanded OAuth session controls.

└──▷ **TRY IT**

Pull aggregated latency, token, and cost statistics for a tracing project to feed into a cost-monitoring dashboard.

```
$ curl -X GET 'https://<langsmith-host>/v2/threads/stats?
project_id=<project_id>' \
  -H 'Authorization: Bearer <api_key>'
```

Set a 14-day lead time on the license expiration banner so self-hosted teams have more runway to renew before the warning appears.

.env

```
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Tag a custom Bedrock model in the LLM Gateway so its traces carry provider metadata and map to provider-specific pricing.

json

```
{
  "model": "my-bedrock-llama3",
  "base_url": "https://bedrock.us-east-1.amazonaws.com",
  "ls_provider": "bedrock"
}
```

› Supports `GET /v2/threads/stats` on self-hosted deployments backed by ClickHouse, including when SmithDB queries are disabled.

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now enforce a download permission check; API callers that previously exported datasets with only read permission will receive a forbidden response.

`snapshot-20260923`

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.

`snapshot-20260922`

LangSmith adds `/v2` /threads/stats, IAM-role ECR support, Gateway fallback timeouts, and `LICENSE_EXPIRATION_WARNING_DAYS` for self-hosted operators.

└──▷ USE IT

Warn the ops team two weeks before a self-hosted LangSmith license expires instead of the default seven days.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Tag a custom OpenAI-compatible model with its provider so the LLM Gateway populates trace metadata and applies the right pricing.

```json
{
  "model": "my-custom-gpt4o",
  "base_url": "https://my-endpoint.example.com/v1",
  "ls_provider": "openai"
}
```

- Adds `GET /v2/threads/stats` endpoint to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project — including on self-hosted ClickHouse-backed deployments.
- Adds `LICENSE_EXPIRATION_WARNING_DAYS` environment variable so self-hosted LangSmith operators can configure how many days before license expiration the warning banner appears (default: 7).
- Adds `ls_provider` field on OpenAI-compatible model configurations in the LLM Gateway to populate trace metadata and match custom models to provider-specific pricing.
- OAuth authorization sessions now accept `owner_type` and `owner_id` parameters, allowing a managed credential to be authorized for either a user or an agent.
- OAuth authorization sessions now accept workspace owners, enabling one managed OAuth credential to be shared across an entire workspace.

+15 more

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read and download permissions; existing API clients that have read-only permission will receive a forbidden response on export requests.

`snapshot-20260921`

└──▷ TRY IT

Pull thread statistics including latency, token usage, and cost for a tracing project to feed into a security posture dashboard or cost audit.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

Give self-hosted operators earlier notice of an approaching license expiration to avoid unexpected access interruptions.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Tag a custom OpenAI-compatible model with its provider so the LLM Gateway records accurate trace metadata and applies provider-specific pricing.

```json
{
  "model": "my-custom-model",
  "ls_provider": "openai",
  "base_url": "https://my-gateway.example.com/v1"
}
```

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read and download permissions; API callers that previously exported with read-only tokens will receive a forbidden response.

snapshot-20260920

LangSmith adds `/v2` /threads/stats endpoint, IAM role-based ECR support, ls_provider tracing, and Gateway fallback timeouts

└──▷ TRY IT

Pull aggregated latency, token, and cost statistics for a tracing project to feed a security-operations dashboard or cost-alerting pipeline.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

Set the license-expiration warning banner to appear 14 days early on a self-hosted deployment so ops teams have more runway before the instance goes dark.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Tag an OpenAI-compatible custom model with `ls_provider` so Gateway traces carry the correct provider label and map to provider-specific pricing.

```json
{
  "model": "my-custom-gpt4",
  "base_url": "https://my-gateway.example.com/v1",
  "ls_provider": "openai"
}
```

- Makes `GET /v2/threads/stats` available on self-hosted deployments backed by ClickHouse, including when SmithDB queries is disabled.
- 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 — users or API tokens that previously exported with only dataset read permission will receive a forbidden response.

`snapshot-20260919`

LangSmith adds `/v2` /threads/stats API, IAM role assumption for ECR , Gateway fallback timeouts, and `LICENSE_EXPIRATION_WARNING_DAYS` config

└──▷ TRY IT

Pull thread-level latency, token usage, and cost statistics for a tracing project in a single API call.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

Route a custom OpenAI-compatible model through the LLM Gateway and tie its traces to provider-specific pricing by setting `ls_provider`.

```json
{
  "model": "my-custom-model",
  "base_url": "https://gateway.langsmith.com/oai/v1",
  "ls_provider": "openai",
  "api_key": "${LANGSMITH_GATEWAY_API_KEY}"
}
```

Set the license-expiration warning horizon to 14 days on a self-hosted deployment so the ops team gets earlier notice.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

- Adds `LICENSE_EXPIRATION_WARNING_DAYS` environment variable for self-hosted operators to configure how many days before license expiration the warning banner appears (default: 7 days).
- Adds `ls_provider` field on OpenAI-compatible model configurations in LLM Gateway to populate trace metadata and match custom models to provider-specific pricing.
- Adds the `GET /v2/threads/stats` endpoint to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project, including on self-hosted ClickHouse-backed deployments.
- Thread query clients can now request server-sent events on `/v2/threads/stats` to receive results progressively while preserving cursor pagination.
- OAuth authorization sessions now accept `owner_type` and `owner_id` parameters, allowing a human to authorize a managed credential for either a user or an agent.

+14 more

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read and download permissions; existing API integrations that rely on dataset read permission alone will receive a forbidden response.

`snapshot-20260918`

LangSmith adds CLI LLM-as-judge evaluator creation, thread evaluator testing, bulk split editing, and new annotation-queue APIs.

└──▷ TRY IT

Add a batch of conversation threads to an annotation queue for human review, mixing thread and run items in one request.

```
$ curl -X POST 'https://api.smith.langchain.com/annotation-queues/<queue_id>/items' \
  -H 'Content-Type: application/json' \
  -H 'x-api-key: <LANGSMITH_API_KEY>' \
  -d '{"items": [{"item_type": "THREAD", "thread_id": "<thread_id>", "session_id": "<session_id>"}, {"item_type": "RUN", "run_id": "<run_id>"}]}'
```

- Adds `langsmith evaluator create-llm` CLI command to define structured LLM-as-judge evaluator rules from a prompt, schema, and model config file, targeting a project or dataset.
- Adds `POST /annotation-queues/<id>/items` endpoint to add RUN items to a single annotation queue; server resolves runs via ClickHouse or SmithDB and returns a standards-shaped items envelope.
- Adds `POST /annotation-queues/<id>/items` support for `item_type: THREAD` (requires `thread_id` + `session_id`) to add conversation threads to an annotation queue; mixed RUN and THREAD batches are supported.
- The `/runs/rules/validate` endpoint now supports thread evaluators — pass `test_thread_id` and `session_id` to test a multi-turn evaluator against a real conversation before saving.
- Adds `evaluator upload --replace` flag to the LangSmith CLI, updating existing code evaluator rules in place and avoiding a delete-before-create window if the replacement upload fails.

+6 more

BREAKING ON UPGRADE

- The legacy feedback formula endpoints (`POST/GET /feedback/formulas` and `GET/PUT/DELETE /feedback/formulas/{feedback_formula_id}`) are deprecated and scheduled for removal on 2026-08-20; migrate existing feedback formulas to composite evaluators before that date.
- Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; use `POST /v2/datasets/<id>/experiment-runs` for paginated experiment comparison going forward.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

01 vLLM

announced · Sep 22, 2026

Local LLM Runtimes

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

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 5/5

vLLM's new vllm-metal brings vLLM's scheduler, paged KV cache, OpenAI-compatible server, structured outputs, LoRA, speculative decoding, and hybrid-attention support to Apple Silicon via MLX and Metal. That makes a serious local serving stack available on Macs while the core v0.30 release also adds Fast Start IPC weight caching, HiSparse KV offload, scale-out endpoints, new model support, and RL sampling-mask returns.

> vLLM shipped faster startup and large-scale serving controls in v0.30.0, led by CUDA IPC weight caching, opt-in scale-out endpoints, new serving flags, watermarking, KV offload, and model-runner improvements. The window also brought vllm-metal support for Apple Silicon serving with MLX/Metal , including memory budgeting, Metal attention kernels, GGUF, speculative decoding, and OpenAI-compatible access.

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

01 IPC weight-cache startup NEW

95

Adds `--load-format ipc_cache` to enable a persistent per-GPU weight-cache daemon that maps post-quantized, TP-sharded weights over CUDA IPC on engine restart. It covers FP4 checkpoints and multi-node TP, avoiding disk reloads on restart.

Restart a vLLM engine instantly by mapping pre-cached weights over CUDA IPC instead of reloading from disk — cuts cold-start time dramatically in multi-restart workflows.

```
$ vllm serve meta-llama/Llama-3.1-8B-Instruct --load-format ipc_cache
```

— Exact flag, command, mechanism, and scope v0.30.0

02 Scale-out endpoint opt-in BREAKING

94

Adds `--enable-scale-out` on `vllm serve` to opt in to scale-out endpoints for prefill/decode separation. This replaces `VLLM_ENABLE_SCALE_OUT_ENDPOINTS`; the environment variable no longer activates scale-out endpoints.

Serve a model with scale-out disaggregated endpoints enabled for prefill/decode separation in large-scale deployments.

```
$ vllm serve deepseek-ai/DeepSeek-V4 --enable-scale-out
```

— Exact replacement and breaking behavior `v0.30.0`

03 **Apple Silicon memory budgeting** NEW 91

Supports `--gpu-memory-utilization` on `vllm serve` to set an inference memory budget on Apple Silicon. The flag caps MLX's reusable buffer cache to prevent accumulation during serving.

Serve a model on Apple Silicon with a controlled memory budget so macOS and other apps retain headroom.

```
$ vllm serve Qwen/Qwen3.5-0.8B --gpu-memory-utilization 0.5
```

— Exact flag with runnable command `launch-20260922-ee3d014c`

04 **vllm-metal MTP speculation** NEW 91

Supports `--speculative-config` with `method:mtp` on `vllm serve` for batched multi-token prediction using an assistant drafter model. It currently supports Gemma 4 with `temperature=0` and `--no-async-scheduling`.

Enable MTP speculative decoding on Gemma 4 to boost throughput under concurrent agent load.

```
$ vllm serve google/gemma-4-E4B-it --gpu-memory-utilization 0.5
--max-model-len 16384 --no-async-scheduling --speculative-config
'{"method":"mtp","model":"mlx-community/gemma-4-E4B-it-
assistant-bf16","num_speculative_tokens":1}'
```

— Exact config, constraints, and command `launch-20260922-ee3d014c`

05 **Multimodal audio backend selection** NEW 89

Adds `audio_backend` in `--media-io-kwargs` for multimodal serving, with torchcodec audio decoding support and soundfile-first automatic decoding. Multimodal hashes also include the `media_io_kwargs` field, with cache hash kwargs scoped by modality.

Serve a multimodal model selecting torchcodec as the audio backend for improved audio decoding performance.

```
$ vllm serve mistralai/Voxtral-Mini-3B-2507 --media-io-kwargs '{"audio_backend": "torchcodec"}'
```

— Exact key, flag, and example command `v0.30.0`

06 **Qwen3.8-Flash-Next offload controls** NEW 83

Adds `--engram-config` to enable UVA PLE offload and Engram tensor parallelism for Qwen3.8-Flash-Next. Adds `VLLM_SPARSE_INDEXER_MAX_LOGITS_MB` to control separate prefill and decode QSA indexer kernel memory limits for Qwen3.8-Flash-Next, and adds DeepSelect top-k for the DSA sparse indexer.

— Exact flag and env var, limited usage detail `v0.30.0`

07 **gRPC server entrypoint change** DEPRECATED 82

Adds `--grpc` to `vllm serve` as the replacement entrypoint for the gRPC server. `python -m vllm.entrypoints.grpc_server` is deprecated in favor of `vllm serve --grpc`.

— Exact old and new commands `v0.30.0`

08 **Apple Silicon vLLM serving** NEW 81

vllm-metal brings vLLM's scheduler, paged KV cache, and OpenAI-compatible server to Apple Silicon via MLX and Metal. It supports pipeline parallelism across multiple Macs over the MLX ring backend; LoRA adapters, structured outputs, separate draft models, prompt-lookup n-gram speculative decoding; experimental vision-language models, text embeddings and reranking, and speech-to-text; and hybrid-attention models from the Qwen3.5, Qwen3.6, Qwen3.8, and Qwen3-Next families.

Query the OpenAI-compatible endpoint exposed by vllm-metal from any client or agent.

```
$ curl http://localhost:8000/v1/chat/completions -H 'Content-Type: application/json' -d '{"model": "Qwen/Qwen3.5-0.8B", "messages": [{"role": "user", "content": "Say hi"}]}'
```

— Many concrete surfaces and curl endpoint `launch-20260922-ee3d014c`

09 **Quantization backend additions** NEW 79

Adds AutoRound 2/3/5/6/7-bit quantization on CUDA, W4A16 DSA with the `nvfp4_fp8_ds_mla` KV cache format, NVFP4 support in the torch linear backend, and FlashInfer CuTeDSL NVFP4 W4A16 as the default over Marlin on SM100/103. Adds `--kv-cache-dtype auto` resolving to `fp8_ds_mla` when FlashMLA is active for DeepSeek V4.

— Exact formats and flag, no example `v0.30.0`

10 **HiSparse KV offload tier** NEW 76

Adds a HiSparse host-resident sparse-MLA decode tier via `HiSparseConnector`. It spills KV pages to pinned host memory under GPU pressure, with per-request GPU hot buffering and Prometheus counters.

— Named connector and behavior, no command `v0.30.0`

11 **Paged Metal attention kernels** NEW 74

Adds a paged varlen Metal attention kernel replacing mlx_lm's padded attention. It enables packed queries with `cu_seqlens` boundaries so prefill and decode run in one mixed model forward across concurrent requests, and automatically uses the NAX attention kernel for compatible prefill batches on M5 Macs.

— Mechanism clear, no user control surface `launch-20260922-ee3d014c`

12 **Removed and deprecated compatibility settings** BREAKING 73

GPTQ activation ordering `g_idx` support has been removed. `VLLM_PREFIX_CACHE_RETENTION_INTERVAL` and `VLLM_MM_HASHER_ALGORITHM`, deprecated in 0.29, have been removed; vendor YaRN aliases no longer re-scale `max_model_len`; and the `all` Mamba cache mode is deprecated.

— Exact removed names and migration impact `v0.30.0`

13 **Targeted online quantization** NEW 72

Adds `quantization_config.targets` to enable targeted online quantization. vLLM also supports partial pre-quantized checkpoints from any quantization method and per-quantization linear backend overrides.

— Config key present, sparse mechanics `v0.30.0`

14 **Sampling mask return flag** NEW 69

Adds `--return-sampling-mask` to return GPU-compacted sampling masks. The release says this fixes an approximately 2x RL step-time regression.

— Exact flag and effect `v0.30.0`

15 **New model and backend support** NEW 67

Adds support for DeepSeek-V4.1-Flash, DeepSeek-V4-Flash-Vision-Exp including ROCm and LoRA, GLM-5.3-Flash, K2-Horizon, Cohere Compass, Bailing V3 VL with MTP, and Nanbeige4.2 via the Transformers backend. Also adds a DeepSeek-V4 CPU backend with AVX512/AMX sparse MLA, indexer, mHC, and compressor kernels, plus K2-Horizon reasoning and tool parsers.

— Specific model list, limited how-to `v0.30.0`

16 **Serving cache and graph improvements** IMPROVED 67

Adds encoder-cache sharing over NIXL and Mooncake for large-scale serving, encoder CUDA graphs for MiniCPM-V 2.5, 2.6, and 4.0, Voxtral Realtime support with `FULL_DECODE_ONLY` graphs, Elastic EP reuse of CUDA graphs across EP reconfiguration, and a KVCR secondary-tier KV cache adapter. Freezing GC during graph capture reduces CUDA graph capture time from 12 s to 2 s and engine init from 28.9 s to 8.2 s on H200.

— Concrete performance data, few controls `v0.30.0`

17 **Gumbel-max watermarked generation** NEW 66

Adds Gumbel-max watermarked generation and detection with a keyed PRF. The feature includes per-request opt-out, an example detection endpoint, and dual-key mode for compatibility with speculative decoding.

— Mechanism named, endpoint not specified `v0.30.0`

18 **Model Runner V2 decoding** NEW 64

Model Runner V2 adds MTP and EAGLE3/DFlash/DSpark speculative decoding under pipeline parallelism, adaptive verification for every draft-model speculator through an online acceptance estimator, and dual-batch overlap in eager mode and with full CUDA graphs for microbatched steps.

— Concrete mechanisms, no invocation surface `v0.30.0`

19 **Hybrid-attention prefix caching** NEW 63

vllm-metal supports prefix caching for hybrid-attention models using vLLM's `align` mode. It saves GDN recurrent state at block boundaries alongside attention KV for multi-turn reuse.

— Mechanism named, no setup details `launch-20260922-ee3d014c`

THINNER COVERAGE BELOW

20 **GGUF on vllm-metal** NEW 45

vllm-metal supports GGUF checkpoints, including Hugging Face config sources for local GGUF weights.

— Named format, thin operational detail `launch-20260922-ee3d014c`

└──▷ BREAKING ON UPGRADE

! Scale-out endpoints are now opt-in via `--enable-scale-out` on `vllm serve`; the `VLLM_ENABLE_SCALE_OUT_ENDPOINTS` environment variable no longer activates them.

! GPTQ activation ordering (`g_idx`) support has been removed.

! The `VLLM_PREFIX_CACHE_RETENTION_INTERVAL` and `VLLM_MM_HASHER_ALGORITHM` environment variables (deprecated in 0.29) have been removed.

! YaRN alignment with Transformers means vendor YaRN aliases no longer re-scale `max_model_len`, which may change effective context lengths for models using those aliases.

! `python -m vllm.entrypoints.grpc_server` is deprecated in favor of `vllm serve --grpc`.

! The `all` Mamba cache mode is deprecated.

announced · Sep 22, 2026

vllm-metal v0.28.0 brings vLLM's scheduler, paged KV cache, and OpenAI-compatible server to Apple Silicon via MLX and Metal.

└──▷ TRY IT

Serve a model on Apple Silicon with a controlled memory budget so macOS and other apps retain headroom.

```
$ vllm serve Qwen/Qwen3.5-0.8B --gpu-memory-utilization 0.5
```

Enable MTP speculative decoding on Gemma 4 to boost throughput under concurrent agent load.

```
$ vllm serve google/gemma-4-E4B-it --gpu-memory-utilization 0.5 --max-model-len 16384 --no-async-scheduling --speculative-config '{"method":"mtp","model":"mlx-community/gemma-4-E4B-it-assistant-bf16","num_speculative_tokens":1}'
```

Query the OpenAI-compatible endpoint exposed by vllm-metal from any client or agent.

```
$ curl http://localhost:8000/v1/chat/completions -H 'Content-Type: application/json' -d '{"model": "Qwen/Qwen3.5-0.8B", "messages": [{"role": "user", "content": "Say hi"}]}'
```

› Supports `--gpu-memory-utilization` flag on `vllm serve` to set an inference memory budget on Apple Silicon, capping MLX's reusable buffer cache to prevent accumulation during serving.

› Supports `--speculative-config` with `method:mtp` on `vllm serve` for batched multi-token prediction (MTP) using an assistant drafter model, currently supporting Gemma 4 with `temperature=0` and `--no-async-scheduling`.

› Adds a paged varlen Metal attention kernel replacing mlx_lm's padded attention, enabling packed queries with `cu_seqlens` boundaries so prefill and decode run in

one mixed model forward across concurrent requests.

- Automatically uses the NAX attention kernel for compatible prefill batches on M5 Macs, leveraging GPU tensor hardware for faster time-to-first-token.
- Supports GGUF checkpoints, including Hugging Face config sources for local GGUF weights.

+5 more

v0.30.0

vLLM v0.30.0 adds Fast Start IPC weight cache, Gumbel-max watermarking, HiSparse KV offload, 8 new models, and opt-in scale-out endpoints.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.30.0 https://github.com/vllm-project/vllm.git
# already have the repo? check out this version:
$ git checkout v0.30.0
```

└──▷ TRY IT

Restart a vLLM engine instantly by mapping pre-cached weights over CUDA IPC instead of reloading from disk — cuts cold-start time dramatically in multi-restart workflows.

```
$ vllm serve meta-llama/Llama-3.1-8B-Instruct --load-format ipc_cache
```

Serve a model with scale-out disaggregated endpoints enabled for prefill/decode separation in large-scale deployments.

```
$ vllm serve deepseek-ai/DeepSeek-V4 --enable-scale-out
```

Serve a multimodal model selecting torchcodec as the audio backend for improved audio decoding performance.

```
$ vllm serve mistralai/Voxtral-Mini-3B-2507 --media-io-kwargs '{"audio_backend": "torchcodec"}'
```

- Adds `--load-format ipc_cache` to enable a persistent per-GPU weight-cache daemon that maps post-quantized, TP-sharded weights over CUDA IPC on engine

restart, covering FP4 checkpoints and multi-node TP — eliminating disk reloads.

› Adds `--return-sampling-mask` flag to return GPU-compacted sampling masks, fixing an approximately 2x RL step-time regression.

› Adds `--engram-config` flag to enable UVA PLE offload and Engram tensor parallelism for Qwen3.8-Flash-Next.

› Adds `--enable-scale-out` flag to `vllm serve` to opt in to scale-out endpoints, replacing the `VLLM_ENABLE_SCALE_OUT_ENDPOINTS` environment variable.

› Adds `--grpc` flag to `vllm serve` as the replacement entrypoint for the gRPC server (deprecating `python -m vllm.entrypoints.grpc_server`).

+26 more

└──▷ BREAKING ON UPGRADE

! Scale-out endpoints are now opt-in via `--enable-scale-out` on `vllm serve`; the `VLLM_ENABLE_SCALE_OUT_ENDPOINTS` environment variable no longer activates them.

! GPTQ activation ordering (`g_idx`) support has been removed.

! The `VLLM_PREFIX_CACHE_RETENTION_INTERVAL` and `VLLM_MM_HASHER_ALGORITHM` environment variables (deprecated in 0.29) have been removed.

! YaRN alignment with Transformers means vendor YaRN aliases no longer re-scale `max_model_len`, which may change effective context lengths for models using those aliases.

! `python -m vllm.entrypoints.grpc_server` is deprecated in favor of `vllm serve --grpc`.

! The `all` Mamba cache mode is deprecated.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

04 Cursor

announced · Sep 23, 2026

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

Cursor now ships PR automation bots that do whole-codebase security review and deployment health monitoring against telemetry, including attack paths, proposed fixes, rollout regression detection, and configurable actions like pausing a rollout or opening a revert PR. This moves coding-agent tooling from code generation into post-PR verification and release operations.

> Cursor shipped two Teams and Enterprise automations: Rollouts for PR-to-production deploy health monitoring and Security Review/Security Reviewer for automated PR vulnerability analysis. Both are enabled from the automations tab and connect pull request review to operational or security findings before code ships.

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

01 Rollouts deploy health monitoring NEW

83

Rollouts is enabled from the dashboard or console automations tab and tracks a pull request from open through production, attaching a monitor to every PR and writing a pre-merge monitoring plan as a PR comment. The plan lists identified risks, intended effects, signals to check, and instrumentation gaps; after deploy, Rollouts compares logs, metrics, and traces against a pre-deploy baseline, tracks health per environment as verified healthy, regression detected, or inconclusive, detects regressions including per-endpoint and per-region, and distinguishes intended effects from regressions. It integrates with Origin or GitHub for source control, continuous delivery systems for deploy events, and Datadog, Grafana, Honeycomb, and other telemetry providers; depending on configuration it can ping the author, pause a progressive rollout, or open a revert PR.

Enable Rollouts to monitor a new deployment and catch regressions per environment as soon as the next PR is opened.

📍 In the Cursor dashboard, go to the automations tab, select 'Rollouts', connect your source control (Origin or GitHub), your continuous delivery system, and your telemetry provider (e.g. Datadog), then save. Rollouts will generate a monitoring plan as a PR comment on the next pull request and track deploy health across staging and production.

Enable Rollouts to monitor a deploy and automatically open a revert PR when a regression is detected.

> 📍 In the Cursor console, go to the automations tab and enable the Rollouts bot, then connect your source control, deploy system, and telemetry provider (Datadog, Grafana, or Honeycomb).

– Includes setup path, integrations, states, and behavior `snapshot-20260924`

`launch-20260923-2f5f6ae2`

---

02 **Security Review PR analysis** NEW 78

Security Review/Security Reviewer is enabled from the dashboard or console automations tab and runs on every non-draft pull request, skipping draft PRs automatically. It performs whole-codebase context analysis and posts one review comment per PR reporting exploitable bugs, covering SQL, command, template, and LDAP injection; missing or broken auth on new and changed routes and auth bypasses; committed secrets and credentials; SSRF; unvalidated redirects; unsafe deserialization; dependency changes introducing known vulnerabilities; vulnerable dependency changes; and insecure infrastructure defaults. Each finding includes severity, attack path, and a proposed fix or one-click fix, and team rules can enforce codebase-specific policies such as permitted external call paths or tables that must not be queried from a request handler on every PR.

Enable Security Review on a repository so every merged PR is automatically checked for exploitable bugs before code ships.

> 📍 In the Cursor dashboard, go to the automations tab, select 'Security Review', choose the repositories you want reviewed, and save. Security Review will post a review comment with severity, attack path, and proposed fix on each subsequent non-draft pull request.

Enable Security Reviewer to automatically audit every PR for vulnerabilities before merge.

> 📍 In the Cursor console, go to the automations tab and enable the Security Reviewer bot.

– Setup path and vulnerability scope are explicit `snapshot-20260924`

`launch-20260923-2f5f6ae2`

└──▷ **ALSO FROM THESE RELEASES**



announced · Sep 23, 2026

Cursor: Enables Rollouts and Security Reviewer bots from the 'automations tab' in Teams and Enterprise plans.

.



└──▷ HOW TO FIND IT

Enable Security Reviewer to automatically audit every PR for vulnerabilities before merge.

> 📍 In the Cursor console, go to the automations tab and enable the Security Reviewer bot.

Enable Rollouts to monitor a deploy and automatically open a revert PR when a regression is detected.

> 📍 In the Cursor console, go to the automations tab and enable the Rollouts bot, then connect your source control, deploy system, and telemetry provider (Datadog,

Grafana, or Honeycomb).

- New Rollouts bot tracks a PR from open through production, writes a pre-merge monitoring plan, detects regressions (including per-endpoint, per-region), distinguishes intended effects from regressions, and can ping the author, pause a progressive rollout, or open a revert PR depending on configuration.
- Integrates Rollouts with Datadog, Grafana, and Honeycomb as telemetry sources for post-deploy signal comparison against a pre-deploy baseline.
- New Security Reviewer bot runs on every PR, performs whole-codebase context analysis for injection (SQL, command, template, LDAP), missing or broken auth on new and changed routes, committed secrets, unsafe deserialization, unvalidated redirects, vulnerable dependency changes, and insecure infrastructure defaults — each finding includes a severity, attack path, and one-click fix.

`snapshot-20260924`

Cursor launches Rollouts (deploy health monitoring) and Security Review (automated PR security analysis) bots for Teams and Enterprise.

.
.
.
.

└──▷ HOW TO FIND IT

Enable Security Review on a repository so every merged PR is automatically checked for exploitable bugs before code ships.

> 📍 In the Cursor dashboard, go to the automations tab, select 'Security Review', choose the repositories you want reviewed, and save. Security Review will post a review comment with severity, attack path, and proposed fix on each subsequent non-draft pull request.

Enable Rollouts to monitor a new deployment and catch regressions per environment as soon as the next PR is opened.

> 📍 In the Cursor dashboard, go to the automations tab, select 'Rollouts', connect your source control (Origin or GitHub), your continuous delivery system, and your telemetry provider (e.g. Datadog), then save. Rollouts will generate a monitoring plan as a PR comment on the next pull request and track deploy health across staging and production.

- Adds Rollouts bot, enabled from the dashboard's automations tab, which attaches a monitor to every pull request, writes a monitoring plan as a PR comment listing identified risks, intended effects, signals to check, and instrumentation gaps, then

tracks deploy health per environment (verified healthy, regression detected, or inconclusive) against logs, metrics, and traces.

- Adds Security Review bot, enabled from the dashboard's automations tab, which posts one review comment per pull request reporting exploitable bugs — covering SQL/command/template injection, auth bypasses, committed secrets and credentials, SSRF, unvalidated redirects, unsafe deserialization, and dependency changes introducing known vulnerabilities — with severity, attack path, and a proposed fix per finding.
- Supports team rules in Security Review to enforce codebase-specific policies (e.g. which external call paths are permitted, which tables must not be queried from a request handler) on every PR.
- Rollouts integrates with Origin or GitHub for source control, continuous delivery systems for deploy events, and Datadog and other telemetry providers for signals.
- Security Review skips draft PRs automatically.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

05 NVIDIA Dynamo v1.5.0

AI Model & Data Infrastructure

Dynamo is a datacenter-scale distributed inference serving framework with disaggregated prefill and decode, KV-cache-aware routing, and multi-engine support.

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 5/5

NVIDIA Dynamo adds a standalone Sidecar that runs beside stock SGLang, TensorRT-LLM, or vLLM engines over their native gRPC APIs, bringing Dynamo routing, KV indexing, TLS/mTLS transport, and pipeline features without rewriting the engine. The release also adds pluggable router policies, tiered KV indexing, a Triton backend for non-LLM models, image/audio/video handling, and new Kubernetes/operator capabilities.

> Dynamo v1.5.0 broadens inference serving with a Triton backend for non-LLM models, an experimental Sidecar for stock engines, pluggable routing, tiered KV indexing, and TLS/ mTLS across request and NATS transports. It also ships Kubernetes/operator changes, multimodal frontend updates, Snapshot separation, GPU memory checkpointing, AISimulate, and several breaking API/runtime migrations.

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

01 AISimulate replaces AIConfigurator

BREAKING

89

AIConfigurator is replaced by AISimulate, shipped as its own `aisimulate` package at v0.12.0 with one CLI. This removes the AIConfigurator wheels and replaces the `dynamo.replay` entrypoint; offline replay now returns a single `ReplayReport` with per-request capture opt-in, so scripts invoking `dynamo.replay` will break.

— Package, entrypoint, report named v1.5.0

02 **Triton backend for Dynamo pipelines** NEW 86

Dynamo adds the `dynamo.triton` backend and `python -m dynamo.triton` launcher to run non-LLM models such as vision, embedding, and tensor-in/tensor-out models inside a Dynamo pipeline alongside LLM workers.

Run a Triton-hosted vision or embedding model inside a Dynamo pipeline without switching to an LLM runtime.

```
$ python -m dynamo.triton
```

— Includes runnable launcher and model scope `v1.5.0`

03 **Snapshot standalone operator and CRDs** BREAKING 84

Dynamo Snapshot moves to a standalone operator and chart, Snapshot v0.1.0, installable separately or through the platform chart with `global.snapshot.install=true` for evaluation. The removed `DynamoCheckpoint` CRD is replaced by `PodSnapshot` CRDs.

— Chart value and CRD migration named `v1.5.0`

04 **TLS across Dynamo transports** NEW 76

Dynamo adds opt-in TLS and mutual TLS on the TCP request/response planes and the NATS transport, with certificate paths injected once through `InfrastructureConfiguration`.

— Names transports and configuration surface `v1.5.0`

05 **Kubernetes v1beta1 storage version** BREAKING 76

`nvidia.com/v1beta1` is promoted to the CRD storage version for Kubernetes, and published examples drop their v1alpha1 copies. Deployments using v1alpha1 examples must migrate because the API server now persists `nvidia.com/v1beta1`.

— Migration impact and version named `v1.5.0`

06 **Multimodal input handling** NEW 70

Dynamo Frontend decodes images once for encode-prefill-decode deployments on SGLang and vLLM using libjpeg-turbo and FFmpeg, then transfers pixels to encode workers over a single NIXL transfer. vLLM workers now accept audio and video inputs, and aggregated workers add realtime audio streaming and dynamic LoRA via vLLM-Omni.

— Mechanism and engines named `v1.5.0`

07 **Sidecar for stock inference engines** NEW 70

Dynamo Sidecar is experimental and runs beside a stock inference engine over its native gRPC API in a separate process. It ships wheel-installed launchers for SGLang, TensorRT-LLM, and vLLM in one CPU-only multi-arch image.

— Architecture and supported engines named v1.5.0

08 **Frontend tokenizer and reasoning controls** NEW 69

Dynamo Frontend adds pluggable tokenizer backends, including a Baseten-contributed backend with a configurable Hugging Face fallback. It also adds an `enable_thinking` gate for Gemma 4 and deployment-level defaults for thinking/reasoning controls, resolved with a single precedence across all chat-template keys.

— Names key and control scope v1.5.0

09 **Pluggable router worker selection** NEW 68

Dynamo Router now supports pluggable worker selection: native scorer and picker policies use the same host-selection path as the built-in selector, while composable worker filters run after eligibility and before scoring. It also adds hard and soft session affinity and a conditional disaggregation bypass that sends short requests straight to decode.

— Mechanism clear, no config names v1.5.0

10 **Typed request rejection errors** IMPROVED 66

Dynamo rejects over-context requests with a typed 4xx before the stream opens; validation failures return HTTP 400 and backend overloads return 503.

— Concrete status codes and behavior v1.5.0

11 **Tiered KV indexing and locality** IMPROVED 65

KV indexing now supports Mooncake shared-cache events for SGLang HiCache without HTTP queries on the hot path, and routes vLLM storage KV events to a disk tier with locality gating. Dynamo also adds video-aware KV hashing for Qwen video inputs.

— Concrete engines, limited usage detail v1.5.0

12 **Rust EPP replaces endpoint picker** BREAKING 60

The Go Endpoint Picker is removed; the Rust EPP is now the default and only endpoint picker, shipped inside the Dynamo Frontend image.

— Before/after clear, no command v1.5.0

13 **SGLang RL rollout contract** NEW 60

Dynamo adds an experimental SGLang RL rollout contract with detailed finish reasons, prompt token IDs and logprobs in `nvext`, native pause controls, and exact stop-token output.

— Fields named, usage absent v1.5.0

14 **Streaming usage on messages API** IMPROVED 60

Dynamo emits per-chunk token usage on `/v1/messages` streaming responses.

— Endpoint and response field behavior named v1.5.0

THINNER COVERAGE BELOW

15 **Operator runtime and canary controls** NEW 59

The Dynamo Operator adds `runtimeVersionOverride`, runtime feature gates, explicit multinode component roles, and default worker canary health checks.

— Names operator fields but little behavior v1.5.0

16 **Batch gateway readiness gating** NEW 53

Dynamo adds an llm-d batch gateway example for offline OpenAI Batch requests through the inference gateway, gated on Dynamo readiness. A new model readiness metric gates batch dispatch.

— Example described, metric unnamed v1.5.0

17 **New vLLM model recipes** NEW 53

Dynamo adds vLLM recipes for Qwen3.8-2.4T-A95B FP8, Qwen3.5-122B FP8 on H200, Nemotron-3.5-Lightning, Nemotron-3-Ultra, and GLM-5.3-Flash.

— Models named, recipe paths absent v1.5.0

18 **KVBM deprecation** DEPRECATED 51

KVBM is deprecated, with removal targeted for v1.6.0; deployments relying on KVBM should migrate before v1.6.0.

— Removal target and migration warning v1.5.0

19 **GPU Memory Service checkpoint lifecycle** NEW 46

Dynamo adds an experimental GPU Memory Service V1 with a checkpoint lifecycle and engine integration for SGLang and vLLM.

— Scope and engines named only v1.5.0

20 **Engine version updates** IMPROVED 46

Dynamo updates engines to SGLang v0.5.18, TensorRT-LLM v1.3.0rc25, and vLLM v0.28.0.

— Exact versions, no behavior v1.5.0

21 **Tool-call streaming and parser coverage** IMPROVED 45

Guided tool calls now stream incrementally by default, and Qwen3 is supported by the v2 unified tool-call parser.

— Behavior named, no usage surface v1.5.0

22 **Native S3 trace sink** NEW 38

Dynamo can write request-trace records to a native S3 sink.

— Sink named, no configuration v1.5.0

23 **Custom ServiceMonitor labels** IMPROVED 33

The Dynamo Helm chart allows custom ServiceMonitor labels.

— Chart area named only v1.5.0

└──▷ BREAKING ON UPGRADE

! `nvidia.com/v1beta1` is now the CRD storage version — the version the API server persists; v1alpha1 copies are removed from published examples and must be migrated.

! The Go Endpoint Picker is removed; the Rust EPP is now the only and default endpoint picker.

! The `DynamoCheckpoint` CRD is removed; `PodSnapshot` CRDs replace it in the standalone Dynamo Snapshot operator.

! AIConfigurator wheels and the `dynamo.replay` entrypoint are replaced by the `aisimulate` package at v0.12.0; existing scripts invoking `dynamo.replay` will break.

! KVBM is deprecated (removal targeted for v1.6.0) — deployments relying on KVBM should migrate before v1.6.0.

v1.5.0

Dynamo v1.5.0 adds pluggable router policies, tiered KV indexing, TLS on all transports, a Triton backend, and a standalone Sidecar for stock inference engines.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.5.0 https://github.com/ai-dynamo/dynamo.git
# already have the repo? check out this version:
$ git checkout v1.5.0
```

└──▷ TRY IT

Run a Triton-hosted vision or embedding model inside a Dynamo pipeline without switching to an LLM runtime.

```
$ python -m dynamo.triton
```

- Adds `runtimeVersionOverride`, runtime feature gates, explicit multinode component roles, and default worker canary health checks to the Dynamo Operator.
- Adds `global.snapshot.install=true` Helm value to pull Dynamo Snapshot into the platform chart for evaluation, with `PodSnapshot` CRDs replacing the removed `DynamoCheckpoint` CRD.
- Supports `python -m dynamo.triton` launcher to bring non-LLM models (vision, embedding, tensor-in/tensor-out) into a Dynamo pipeline alongside LLM workers via the new `dynamo.triton` backend.
- Adds opt-in TLS and mutual TLS on the TCP request/response planes and the NATS transport, with certificate paths injected once through `InfrastructureConfiguration`.
- Introduces pluggable worker selection: native scorer and picker policies run through the same host-selection path as the built-in selector, with composable worker filters applying after eligibility and before scoring.

+24 more

BREAKING ON UPGRADE

- `nvidia.com/v1beta1` is now the CRD storage version — the version the API server persists; v1alpha1 copies are removed from published examples and must be migrated.
- The Go Endpoint Picker is removed; the Rust EPP is now the only and default endpoint picker.
- The `DynamoCheckpoint` CRD is removed; `PodSnapshot` CRDs replace it in the standalone Dynamo Snapshot operator.
- AIConfigurator wheels and the `dynamo.replay` entrypoint are replaced by the `aisimulate` package at v0.12.0; existing scripts invoking `dynamo.replay` will break.
- KVBM is deprecated (removal targeted for v1.6.0) — deployments relying on KVBM should migrate before v1.6.0.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

// END OF TRANSMISSION

The Cyber Toolchain · This Week's Highlights · September 25, 2026

The full firehose — every feature release, every day — is in the newsletter.

▷ Subscribe