NOW AVAILABLE The 512GB Mac Studio is on sale now, shipping February to March. See every configuration.

THE AI TOOLCHAIN

NO. 083

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED NOVEMBER 4, 2026 · EVERY DAY

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

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

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 357 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 87 of 89 tools

▾ // THE BRIEF

HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: Opus 5 | GPT-5.5

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

vim ? help

Three unrelated things stand out. Qdrant 1.20's TurboQuant datatypes shrink vectors without paying the usual rescoring pass back in latency. MLflow 3.18.0 adds online evaluation and statistical regression comparison, so agent quality changes are measured rather than argued about. Amp's service accounts give CI-triggered threads a workspace identity instead of borrowing an employee's key.

DEPLOY

Store more vectors per node without trading recall back for a rescoring pass

TurboQuant-based quantization datatypes cut vector storage with no rescoring penalty, and nested payload fields now get per-element indexing. The usual quantization bargain — compress, then re-read full-precision vectors to fix the ranking — is what made the memory savings cost latency; filters over nested arrays also stop degrading into scans.

Qdrant

EVALUATE

Catch an agent quality regression with a statistical test instead of eyeballing a diff

3.18.0 ships automatic issue detection, online evaluation and statistical regression comparison for GenAI agents, plus a skill registry. Comparing two agent versions by reading sample outputs is how real regressions ship; a significance test on scored runs gives a release decision someone can defend.

MLflow

GOVERN

Make automated runs belong to the org, not to whoever's key was handy

Amp's service accounts are workspace-owned identities with their own API keys, so CI-triggered threads survive the engineer who created them leaving. Claude Code adds credential expiry to `auth status` and headless JSON output, plus gateway role assumption so AWS spend attributes to the right team.

Amp · Claude Code

OPERATE

Stop attributing LLM spend from a stale price table

> Phoenix refreshes its built-in pricing catalog roughly daily with no server restart, so cost dashboards stop drifting as providers reprice. litellm-rs now carries cost in usage data with OpenRouter settling from provider-reported cost, and traccia-py adds a diagnostic for why a call failed to match a price.
>
> Arize Phoenix · litellm-rs

Does **Opus 5** read better? 👍 👎

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.177 NOTES ↗ RANK

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

> xalgorix hardened its scan-completion and proof logic to stop premature or unverifiable closures, added a configurable idle/verification window with partial-completion reporting, and extended its LLM provider catalog with Runcrate and cross-provider cost tracking, plus an arjun flag auto-correction fix.

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

01 Configurable scan convergence and idle-tail window NEW

90

Control how many idle iterations a scan waits before finishing with `XALGORIX_CONVERGE_FINISH_IDLE` (default 500, up from 150), and grant active high-confidence leads a bounded verification window via `XALGORIX_CONVERGE_FINISH_LEAD_WINDOW` (default 100 iterations) before they are closed. Both can be edited in the web UI under Environment → Runtime without restarting.

— Named env vars with exact defaults and a UI path to edit them. v4.6.175

02 Arjun flag auto-correction IMPROVED

75

xalgorix now maps arjun options that agents commonly guess — `--threads`, `--chunk`, `--delay`, `--stable-sleep`, `--timeout`, `-oJ` — to arjun's real `-t` and `--rate-limit` flags, and limits rate-limit rewriting to commands that actually invoke a tool.

— Full list of guessed and real flags named, mechanism clear. v4.6.175

03 Ledger and verification integrity improvements IMPROVED

65

xalgorix now keeps confirmed SQL injection, template injection, and in-band XXE findings — verifiers mark their confirmations proven, including when a confirmation merges into an existing ledger entry, so a scan can no longer end with a confirmed injection unreported; treats out-of-band and cross-identity signals as never auto-closeable, keeping them open for the model to conclude while the verification window waits on them; restricts ledger proof to engine-verified evidence only, excluding evidence the model writes itself (including text such as not confirmed), and records which engine verifier wrote each piece of evidence; enforces per-type severity caps after CVSS scoring so a model-supplied vector can no longer raise capped finding types such as clickjacking or missing headers to medium or high; and keeps path case in ledger

and finding-dedup keys — only the scheme and host are lowercased — so a case-variant route such as `/Api/...` is no longer merged into `/api/...` and a case-variant authentication bypass is its own finding.

— Internal verification mechanism explained but no reader action. Clear mechanism, no named config surface for readers to use. Mechanism well described but purely internal to the engine. Names capped finding types but no reader-facing control. Concrete before/after example of dedup key behavior.

v4.6.177

---

04 **Partial-completion reporting for unfinished scans** IMPROVED

65

Scans that close work unexecuted and find nothing now report `completion: partial` with `stop_reason: untested_at_finish`, and unrun tasks are marked `untested_at_finish` rather than not applicable, with open leads listed in finish telemetry.

— Named status fields readers can check in scan output. v4.6.175

---

05 **Cross-provider LLM cost reporting** NEW

60

xalgorix now adds up LLM charges reported by providers (such as OpenRouter's `usage.cost`, including bring-your-own-key upstream charges) across the scan and all subagents, and displays the total in the CLI's end-of-scan usage note.

— Names the field and where output appears, actionable to check. v4.6.176

THINNER COVERAGE BELOW

06 **Runcrate added as built-in LLM provider** NEW

55

xalgorix adds Runcrate — a chat-completions API for open-weight models — to the built-in provider catalog, with model discovery via `/v1/models`.

— Names the provider and discovery endpoint, light on further detail. v4.6.176

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

yetone magpie

Feedback ⓘ v0.1.2914 CODE ↗ RANK

Magpie is a macOS menu-bar app, TUI, and CLI that lists every AI coding agent on the machine and switches each one's model, such as Codex on DeepSeek or Claude Code on Kimi, by rewriting only that key in the agent's own config.

> Magpie's biggest update lets each AI agent carry its own conversation-compaction window instead of sharing one global setting, alongside new image generation support for GitHub Copilot accounts and OTEL tagging for multi-server deployments.

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

01 **Per-agent conversation compaction windows** NEW 95

The `magpie model compact <agent>` CLI command sets per-agent Long conversations compaction windows for Codex, Claude, and Grok Build, backed by `settings.AgentCompacts` and the `SetAgentCompact` function, so agents whose models differ no longer share a single global compact threshold. A `POST /api/settings/agent-compact` endpoint (also referenced as `settings/agent-compact`) reads and writes each agent's own Long conversations size or full-context mode independently of the global setting.

Give Codex its own compaction threshold without affecting Claude or Grok's settings.

```
$ magpie model compact codex 300k
```

Let Grok run all the way to its full context window, overriding any global compaction setting.

```
$ magpie model compact grok off
```

Remove a previously set per-agent compaction window for Codex, reverting it to follow the global setting.

```
$ magpie model compact codex every
```

Pin Codex to a 600 K-token compact window while leaving every other agent at the global 400 K default.

```go
SetAgentCompact("codex", &settings.AgentCompact{At: 600000})
SetCompactAt(400000)
```

Put Claude Code into full-context mode (no compaction) via the API while other agents keep their own windows.

```
$ curl -X POST http://localhost:<port>/settings/agent-compact -d
'{"agent": "claude", "full": true}'
```

— Names CLI command, config key, API endpoint and runnable examples v0.1.2914 v0.1.2910

02 **OTEL resource attributes for multi-server tracing** NEW 90

`OTEL_RESOURCE_ATTRIBUTES` and `OTEL_SERVICE_NAME` environment variables are now read and merged into every OTLP export's resource attributes, letting traces and metrics be tagged with the originating server, with `OTEL_SERVICE_NAME` taking highest precedence over any `service.name` in `OTEL_RESOURCE_ATTRIBUTES`.

Distinguish traces and metrics from multiple servers sending to the same OTLP collector by labelling each server with its own service name and environment.

```
$ OTEL_SERVICE_NAME=api-gateway
OTEL_RESOURCE_ATTRIBUTES=deployment.environment=production,host.r
egion=us-east-1 magpie
```

— Names exact env vars, precedence rule, and a runnable example v0.1.2900

03 **Image generation for GitHub Copilot accounts** NEW 80

GitHub Copilot accounts now surface their image models (such as GPT Image 2.5 Flare) discovered from the account's `/models` endpoint, drawing at `/images/generations` and `/images/edits` — mirroring how Copilot CLI's `image_generation` tool works — while keeping those models out of agent chat.

— Names endpoints and model but no CLI usage shown v0.1.2889

THINNER COVERAGE BELOW

04 **Per-word Masked words and Never mask lists** IMPROVED 55

The Privacy settings panel now shows Masked words and Never mask as per-row lists, with each word on its own line, editable in place and saved on Enter or blur, replacing a single comma-separated box.

— Names UI panel and interaction but no config key given v0.1.2896

05 **Usage windows for remote provider accounts** NEW 50

Remote magpie provider connections now surface usage windows for Claude and Codex accounts in the Accounts and keys view, with per-account rows when the remote holds more than one account and a clear error state when the remote's data could not be read.

— Describes UI behaviour but no named flag or endpoint v0.1.2908

NEW INTEGRATION DeepSeek Harness project skills support

v0.1.2901 CODE ↗ STABLE

magpie now links project skills for DeepSeek Harness into `.agents/skills` at the project root, determined by the nearest `.git` ancestor folder.

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.2901
```

git

```
$ git clone --branch v0.1.2901 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.2901
```

published under the project's GitHub account · checked 4 Nov Wrong command?

magpie now supports DeepSeek Harness (`dsh`) in its project skills directory map, linking shared skills into `<projectRoot>/.agents/skills` and adding `/.agents/skills/<skill>` to `.gitignore` automatically.

WAS THIS USEFUL? 👍 👎

zot

Feedback ⓘ v0.4.62 NOTES ↗ RANK

zot manages coding agent workflows from the command line.

> zot's single release this window adds OpenRouter usage attribution, human-readable session aliases for resuming work by name, and refreshed Anthropic/OpenAI model catalogs.

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

01 **OpenRouter app-attribution headers** NEW 45

zot now automatically sends app-attribution headers when using the OpenRouter provider, enabling per-app usage tracking on the OpenRouter dashboard.

— Explains purpose but no header names or config shown v0.4.62

02 **Readable session aliases and named startup** NEW 40

zot sessions now support readable aliases and named session startup, letting you reference and resume sessions by name instead of opaque IDs.

— Describes capability but no exact command syntax given v0.4.62

03 **Updated Anthropic and OpenAI model listings** IMPROVED 25

zot now supports current Anthropic and OpenAI model listings, keeping the built-in provider model catalogs up to date.

— Bare statement of catalog update, no model names listed v0.4.62

WAS THIS USEFUL? 👍 👎

aerovato Operator Memory

Feedback ⓘ deepseek@v0.3.0

NOTES ↗ RANK

Operator Memory is a self-improving context engine for coding agents.

> Operator Memory extended harness support to DeepSeek and OpenCode V2, rolled its machine-stable versioned preamble out further (now including the Pi harness), and added internal tooling to coordinate multi-package releases.

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

01 **Harness support for DeepSeek and OpenCode V2** NEW 70

Operator Memory can now be installed into a DeepSeek agent environment with `operator-helper install deepseek` and into an OpenCode V2 project with `operator-helper install opencode`, both described as fully supported harnesses that retain context across sessions.

Add Operator Memory support to a DeepSeek agent environment for the first time.

```
$ operator-helper install deepseek
```

Add Operator Memory to an OpenCode V2 project so the agent retains context across sessions.

```
$ operator-helper install opencode
```

– Exact install commands given for both harnesses deepseek@v0.3.0 opencode-v2@v2.3.0

THINNER COVERAGE BELOW

02 **Machine-stable versioned preamble for deterministic context loading**

IMPROVED 35

Operator Memory rolls out a machine-stable versioned preamble to keep context loading deterministic across sessions, with this rollout extended to the Pi harness.

– Purpose stated but no mechanism or command detail deepseek@v0.3.0 pi@v1.2.0

03 **Coordinated package publishing script** NEW 20

Operator Memory adds a coordinated package publishing script to synchronise releases across the multi-package repository.

— Internal tooling described only in generic prose `pi@v1.2.0`

WAS THIS USEFUL?  

Alibaba Qwen Code

Feedback ⓘ v0.26.7 NOTES ↗ RANK

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

> Qwen Code v0.26.7 adds remote-access control for `qwen serve` with token authentication, a Docker-based `/verify-pr` sandbox for running untrusted PR code, several Web Shell/GitHub channel improvements, and concurrent workspace uploads.

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

01 Remote connections to `qwen serve` NEW

90

Qwen Code now supports enabling remote connections to a running `qwen serve` instance, including bearer-token authentication (e.g. `qwen serve --require-auth --open` with the `QWEN_SERVER_TOKEN` environment variable) and client disconnects without restarting the server, configurable via Settings → Connections → Allow remote connections.

Allow a remote client or CI runner to connect to a running `qwen serve` instance with bearer-token authentication without restarting the daemon.

```
$ QWEN_SERVER_TOKEN=mysecrettoken qwen serve --require-auth --
open
```

— Named settings path, env var, and a runnable command. v0.26.7

02 Docker sandbox for untrusted PR code via `/verify-pr` NEW

75

A local `/verify-pr` Docker recipe lets you run untrusted PR code safely, proxying model credentials through a chat-completion proxy running on the host so credentials don't enter the sandbox.

— Named command and proxy mechanism; no run instructions given. v0.26.7

03 `prompt.submit` hooks run before prompt processing IMPROVED

70

Run Mods `prompt.submit` hooks now execute in the Ink terminal before typed prompts are processed, so hooks see composer text with pasted content already expanded.

— Named hook and timing behavior; no usage example provided. v0.26.7

04 Channel events page in Web Shell NEW

60

Web Shell adds a Channel events page providing a filterable, auto-refreshing timeline of Channel activity, including skip reasons, replies, task transitions, and poll results.

— UI location named; no exact navigation path given. v0.26.7

05 **GitHub settings editor in Web Shell channel editor** IMPROVED

60

The Web Shell channel editor now lets you edit GitHub repository access, mention behavior, and notification polling settings directly in the UI instead of requiring manual settings changes.

— UI capability named; no exact settings path given. v0.26.7

THINNER COVERAGE BELOW

06 **GitHub events routed to bound PR sessions** NEW

50

GitHub events for bound PRs now route to the associated session as untrusted prompts, with replies staying private unless explicitly published.

— Behavior described; no config or endpoint named. v0.26.7

07 **Concurrent workspace uploads** IMPROVED

50

Workspace uploads now proceed concurrently while still rejecting occupied destinations and preserving directory occupancy protections.

— Behavior described; no config or limits specified. v0.26.7

08 **Model configuration at session creation in managed-agent API**

IMPROVED

30

The managed-agent API now supports specifying full model configuration when a session is created.

— No endpoint, parameters, or example given. v0.26.7

└──▷ ALSO FROM THESE RELEASES

Prevent background model-catalog network calls in air-gapped or privacy-sensitive environments while keeping locally cached metadata available.

yaml

```yaml
privacy:
  usageStatisticsEnabled: false
```

WAS THIS USEFUL? 👍 👎

coffee-cli Feedback ⓘ v3.8.8 NOTES ↗ RANK

coffee-cli is a clean AI terminal workspace with quick input, folder switching, copy & paste, and image pasting.

> Coffee CLI's v3.8.7 release focuses on agent experience: the Coffee Agent now accumulates and reuses task-specific knowledge, replies show per-tool usage, and screenshots can be pasted directly into AI prompts across multiple coding assistants.

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

01 **Agent experience memory** NEW 55

Coffee Agent now accumulates project-specific experience: when a task succeeds, it records the approach and reuses it for similar tasks; when memory is nearly full it merges old notes instead of dropping them.

– Explains mechanism but no config surface to invoke it v3.8.7

02 **Taskbar flash alerts for AI tasks** IMPROVED 55

The taskbar button now flashes continuously when an AI is waiting for approval, and flashes once when a task completes, working even when sounds are turned off.

– Describes exact trigger conditions for the alert v3.8.7

03 **Screenshot paste in answer boxes** NEW 55

Claude Code, Codex, and DeepSeek Harness answer boxes now accept pasted screenshots directly into AI prompts.

– Names three target integrations but no further mechanism detail v3.8.7

04 **Per-reply tool usage visibility** NEW 50

Each Agent reply now lists the tools it used, with failures marked, and the list is expandable on click.

– Clear UI interaction described, no deeper mechanism v3.8.7

05 **Pinned active sessions in history** IMPROVED 35

Active sessions now stay pinned to the top of the history list in tab order instead of jumping around.

– Minor UI fix with limited detail on scope v3.8.7

NEW INTEGRATION Antigravity CLI conversation history

v3.8.8 NOTES ↗ LATEST STABLE

Coffee CLI adds per-card folder memory, a `+` tab button, local settings storage for credentials, Coffee Agent UI improvements, and larger text across the app.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v3.8.8 https://github.com/edison7009/Coffee-CLI.git
# already have the repo? check out this version:
$ git checkout v3.8.8
```

Coffee CLI now stores model keys, message-channel tokens, and SSH passwords in local settings files instead of the system keychain, eliminating keychain prompts on every launch — re-enter credentials once after updating.

Each tool card now remembers the last folder it was opened in, so one click resumes where you left off; use the folder icon to switch directories.

A new `+` button at the right of the tab bar now works the same as `New session`.

Coffee Agent now shows a `New chat` button beside the input box and displays the Coffee CLI icon in the sidebar.

Claude Code's desktop view now shows a live status line of the current step with animated transitions between steps.

+1 more

└──▷ **BREAKING ON UPGRADE**

! Model keys, message-channel tokens, and SSH passwords are moved from the system keychain to local settings files; all credentials must be re-entered once after updating.

WAS THIS USEFUL? 👍 👎

pstack-claude

Feedback ⓘ v0.9.128 NOTES ↗ RANK

pstack-claude is a port of Poteto's pstack agent framework to Claude Code, Codex, Copilot, Pi, and other AI coding assistants.

> pstack-claude's biggest window addition is MiniMax Code platform support, alongside a full redesign of the Orchestrate playbook around coordinator-native tools and a new `pi agent tools: off` delegation mode; several breaking changes also land, including a watcher failure-kind rename, dropped proxy forwarding to the statechart runner, and removal of Codex prompt stubs.

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

01 **MiniMax Code platform support** NEW 100

pstack-claude now runs on MiniMax Code 0.7.4, loading all 67 skills and 4 agents through MiniMax's Claude Code reader with no separate manifest — install by copying `plugins/pstack` into MiniMax's `plugins/` data directory. The `SessionStart` hook recognizes MiniMax via the `MINIMAX_PLUGIN_ROOT` environment variable, injects the poteto-mode mandate in MiniMax wording, and reads session configuration from `~/.minimax/pstack-models.md`. A new `minimax-tools.md` at `plugins/pstack/skills/poteto-mode/references/minimax-tools.md` maps Claude tools to MiniMax equivalents using MiniMax Code 0.7.4 tool schemas, and the plugin generator now fails when a top-level `plugin.json` or `.minimax-plugin/plugin.json` already exists, preventing silent override of MiniMax's loader precedence over `.claude-plugin/plugin.json`.

— Full mechanism, named files, and an exact install step. v0.9.125

02 **Orchestrate playbook redesign with coordinator-native tools** 90

BREAKING

The Orchestrate playbook now runs with the coordinator's own tools — the Agent tool for workers, a store under `~/.claude/orchestrate/<slug>/` with `notes.md` as its tracker, and PR and CI event subscriptions — replacing scripted polling loops. Two sections, Topology and Supervision, replace all nine earlier sections from Roles and placement through Escalation, dropping the `preferences.md` register and the rule for preserving human rulings verbatim. New substitution rules rewrite the Project tracker and Project store sentences, the cloud VM sentence, a control-skill check, and the forking-a-worker line, with the denylist now rejecting Project tracker and Project store. The playbook no longer references `units.tsv`, `frontier.json`, or the inbox from stores written by earlier releases, and `orch/orch.ts`, `orch/store.ts`, `orch/orch.test.ts` are deleted along with the `orch` subcommand, its `test` script, and `tla/OrchStoreLock`.

— Detailed before/after but mostly descriptive, not a command. v0.9.123

03 **Environment isolation drops proxy forwarding to statechart runner**

90

BREAKING

Child processes no longer inherit the whole environment, so `HTTPS_PROXY` is no longer forwarded to the statechart runner for mise and Java; proxy-dependent `mise install` runs (e.g. for TLC or Java) will fail unless `TLA2TOOLS` and `JAVA` are set directly or `mise install` is run separately with the proxy configured.

— Exact env vars and remediation command given. v0.9.128

04 **pi agent tools: off delegation mode** NEW

80

A new `pi agent tools: off` setting delegates all agent tools to another Pi extension such as `pi-subagents`. When set, the `pstack-pi-tools` system prompt section no longer adds the sentence instructing the model to delegate with the session's own subagent tool, and the tool list sheet line removes `agent`, `send_message`, `list_agents`, `stop_agent`, and `/pstack-agents`.

— Names exact setting and tools removed, clear config to flip. v0.9.127 v0.9.128

05 **Codex prompt stubs removed** BREAKING

75

The 43 Codex prompt stubs under `plugins/pstack/.codex-plugin/prompts/` are removed; Codex 0.166.0 no longer loads them via plugin install or the `~/.codex/prompts` recipe, and the corresponding documentation in `docs/reference.md` is gone. The stub generator no longer writes stubs, and a `commands/` folder will still cause it to fail.

— Names exact paths and failure condition but little reader action. v0.9.127

THINNER COVERAGE BELOW

06 **Per-agent-type descriptions in subagent_type parameter**

IMPROVED

55

The `agent` tool's `subagent_type` description now lists each type with its `description:` field from the agent file — including YAML `|` and `>` block scalars — so a model can distinguish when to use each type from `pstack-agents`.

— Names the parameter and YAML scalar handling but is automatic, not configurable.

v0.9.121

07 **Watcher query failure kind renamed** BREAKING

50

The watcher's `missing-key` query failure kind is renamed to `missing-field`; any consumer that pattern-matches on `missing-key` will stop matching and must be

updated.

— Clear rename but no migration mechanism beyond the name. `v0.9.128`

└──▷ BREAKING ON UPGRADE

! The watcher's `missing-key` query failure kind is renamed to `missing-field`; any consumer that pattern-matches on `missing-key` will stop matching and must be updated.

! No child process inherits the whole environment; `HTTPS_PROXY` is no longer forwarded to the statechart runner for mise and Java, so proxy-dependent `mise install` runs (e.g. for TLC or Java) will fail unless `TLA2TOOLS` and `JAVA` are set explicitly or `mise install` is run separately with the proxy configured.

! The 43 Codex prompt stubs under `plugins/pstack/.codex-plugin/prompts/` are removed; Codex 0.166.0 no longer loads them via plugin install or the `~/.codex/prompts` recipe, and the documentation in `docs/reference.md` for that recipe is gone. The stub generator no longer writes stubs, and a `commands/` folder will still cause it to fail.

! The plugin generator fails (instead of proceeding) when a top-level `plugin.json` or `.minimax-plugin/plugin.json` exists in the plugin directory.

! `orch/orch.ts`, `orch/store.ts`, and `orch/orch.test.ts` are deleted; the `orch` subcommand and its `test` script are removed, and `tla/OrchStoreLock` is gone.

! The `preferences.md` register of standing orders is removed, and the playbook no longer reads `units.tsv`, `frontier.json`, or the inbox from stores written by earlier releases.

WAS THIS USEFUL?  

Anthropic Claude Code

Feedback ⓘ v2.1.318 NOTES ↗

CODE ↗ RANK

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

> Claude Code adds credential expiry visibility to auth status and headless JSON output, mod fork effort control, gateway role assumption for AWS cost attribution, and runner/connector URL blocking, alongside breaking changes to self-hosted runner log output and IDE extension installation.

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

01 Self-hosted runner log format changes

BREAKING 83

`ok session=…` now prints at every log level on stdout (previously debug-only on stderr) and adds `reason=…` after `jti=…`; `nack session=…` now ends with `reason=…` after `retryable=…`. Scripts matching `retryable=true:` or `retryable=false:` need updating.

— Exact log formats named with migration note v2.1.318

02 Credential expiry visibility in auth status and headless mode

NEW 78

`claude auth status` now reports a new `loginExpiresAt` field, and `claude -p` JSON output includes a new `credential_expires_at` field, showing when a stored claude.ai sign-in expires. `claude -p` text output also prints a one-line stderr warning starting 10 days before a stored sign-in ends.

Run a headless task and capture credential expiry alongside results so a CI wrapper can alert before the token lapses.

```
$ claude -p 'summarize recent commits' --output-format json | jq '{summary: .result, expires: .credential_expires_at}'
```

— Named fields and example command given, no further mechanism v2.1.318

03 Per-account role assumption in gateway config

NEW 70

Add `assume_role` to `mantle` upstreams in the Claude apps gateway config to assume a role in another AWS account, enabling per-developer AWS cost attribution.

— Config key named but no file path or example v2.1.318

04 **Spawn-reason context for runner orchestrator hooks** IMPROVED

70

Self-hosted runner orchestrator spawn-reason hooks now receive a `CLAUDE_RUNNER_SPAWN_REASON` environment variable explaining why Anthropic sent the request, with example values like `runner_lost` or `resumed`; it is unset when no reason was sent.

— Env var and example values named, no code sample v2.1.318

05 **URL-based blocking for MCP servers and connectors** IMPROVED

70

Block Anthropic-provided servers and cloud/runner connectors by URL using `deniedMcpServers` URL entries, which now also match a connector's own URL in cloud sessions and on self-hosted runners.

— Config key named, behavior change explained v2.1.318

THINNER COVERAGE BELOW

06 **Effort control for mod forks** NEW

55

Set an `effort` option on `$.model.fork` to control the effort level of a mod's side job — for example set it to `low` regardless of what effort the main session uses. The result reports the effort that was actually sent.

— Config key named, no example command shown v2.1.318

07 **IDE extension auto-install removed from IDE terminal** BREAKING

55

Claude Code no longer installs or updates the IDE extension when started in an IDE terminal; use `/ide` from an outside terminal or your editor's extensions view instead.

— Clear migration path given, limited mechanism detail v2.1.318

08 **No-repository option for cloud sessions** NEW

35

Start a cloud session without picking a repository using the new No repository option in Anthropic-hosted environments.

— UI option named, no further mechanism v2.1.318

└──▷ **BREAKING ON UPGRADE**

! Self-hosted runner log format changed: `ok session=…` now prints at every log level on stdout (was debug-only on stderr), adds `reason=…` after `jti=…`, and `nack session=…` ends with `reason=…` after `retryable=…` — update any scripts that match `retryable=true:` or `retryable=false:`.

! Claude Code no longer installs or updates the IDE extension when started in an IDE terminal; use `/ide` from an outside terminal or your editor's extensions view instead.

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.102 NOTES ↗

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

Graphify adds `--depth` flag and `query_budget` / `query_depth` config keys to `graphify query`, and makes `graphify update` re-extract only changed files.

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install graphifyy==0.9.102
# already installed? upgrade to this version:
$ pipx install --force graphifyy==0.9.102
```

git

```
$ git clone --branch v0.9.102 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.102
```

PyPI attestation · checked 4 Nov Wrong command?

Control query scope by setting `query_budget` and `query_depth` in `.graphifyrc` — `graphify query` now reads both keys to cap traversal cost and depth.

Limit query traversal depth per-invocation with the new `--depth` flag on `graphify query`.

`graphify update` now re-extracts only files whose content changed, producing a graph identical to a full rebuild, so incremental refreshes are safe to use in CI.

- UPDATED SUPPORTED PLATFORM Go call-resolution improvements
- UPDATED SUPPORTED PLATFORM Rust call-resolution improvements
- UPDATED SUPPORTED PLATFORM Kotlin call-resolution improvements
- UPDATED SUPPORTED PLATFORM Python call-resolution improvements
- UPDATED SUPPORTED PLATFORM JavaScript/TypeScript call-resolution improvements
- UPDATED SUPPORTED PLATFORM Solidity import resolution
- UPDATED SUPPORTED PLATFORM C++ call-resolution improvements

└──▷ BREAKING ON UPGRADE

! The AST cache schema is bumped to version 37; warm caches will re-run affected extractions on first use.

WAS THIS USEFUL? 👍 👎

martin-loop

Feedback ⓘ v0.9.0 NOTES ↗ RANK

Run coding agents without babysitting them. Keep jobs focused, bounded, checked and accountable from start to finish.

> martin-loop v0.9.0 tightens governed Swarm execution with mandatory agent/concurrency and budget-authority rules, and adds MCP client controls to pause, continue, and cancel local runs with hardened auditability.

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

01 **MCP run control: pause, continue, cancel** NEW 75

MCP clients can now pause, continue, and cancel eligible local runs through capability-aware controls, with terminal-state precedence and auditable receipts recording requested-versus-applied outcomes, plus exact cancellation signal/publication binding. The control plane is hardened with control journals and execution leases that fail closed on replay, link, stale-lock, PID-reuse, and concurrent-reclaimer scenarios.

— Describes capability and hardening mechanisms, no exact command or endpoint named

mcp-v0.9.0 v0.9.0

02 **Governed Swarm execution count and budget requirements** 65

BREAKING

Governed Swarm execution now requires explicit agent count and concurrency to be specified; governed runs that omit these will fail. Runs must also declare budget authority — provider-free runs require exactly $0 and live runs require a positive budget authority value — which binds successful evidence to canonical provider settlement.

— Explains mechanism and before/after but no exact config keys or flags given v0.9.0

└──▷ **BREAKING ON UPGRADE**

! Governed Swarm execution now requires explicit agent count and concurrency — existing governed runs that did not specify these will fail.

! Provider-free governed runs now require exactly $0 budget authority; live runs require a positive budget authority value.

WAS THIS USEFUL? 👍 👎

All Hands AI OpenHands

Feedback ⓘ v1.35.0 NOTES ↗

RANK

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

> OpenHands' v1.35.0 release is mostly workflow polish: deeper automation configuration (keep_alive editing, webhook event sources, template suggestions, Bitbucket Data Center clone URLs), a new Files tab download button, chat status/grouping improvements, and sidebar layout persistence.

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

01 **Automation configuration and webhook enhancements** IMPROVED 70

Automations gained several configuration surfaces in one update: the automation detail page now shows and lets you edit `keep_alive`; a new Events webhook sources view under Automations lets you browse and manage incoming webhook event sources, each listing its own event types so triggers can be configured more precisely; automation template suggestions now match an instance's configured integrations; and Bitbucket Data Center repositories picked in automations are stored as clone URLs for reliable re-use.

— Names a config field and UI areas but no API or CLI surface. v1.35.0

THINNER COVERAGE BELOW

02 **Sidebar layout persistence and resizing** IMPROVED 50

Sidebar enhancements store sidebar pin and archive state as server-side tags so the layout persists across sessions, show startup activity in sidebar conversation rows, and allow dragging the sidebar edge to resize it.

— UI-only description; drag-to-resize is directly actionable. v1.35.0

03 **Chat UI live status and message grouping** IMPROVED 45

Chat UI enhancements add a live agent status display, compact message timestamps shown on hover, and grouping of hooks and reasoning with tool calls along with a summary of what each group did.

— Describes UI behavior without configuration or command details. v1.35.0

04 **OHE members routed to MCP settings** NEW 45

OHE members are sent to Settings > MCP when organization MCP servers are enabled.

— Exact navigation path given but no further mechanism. v1.35.0

05 **Download button in Files tab** NEW

40

A Download button in the Files tab lets users save workspace files directly from the browser.

— Clear UI location but no format or size details. v1.35.0

06 **Instant conversation open with startup progress** IMPROVED

40

Conversations now open immediately with inline startup progress shown instead of waiting for runtime readiness.

— Behavioral change described without technical mechanism. v1.35.0

07 **Persisted cloud repository selection on home page** IMPROVED

35

The home page now remembers and allows clearing the selected cloud repository.

— Simple UI behavior described with no further detail. v1.35.0

08 **Workspace and repository metadata as conversation tags** NEW

30

Workspace and repository metadata is now persisted as server-side conversation tags.

— No schema or retrieval mechanism given for the tags. v1.35.0

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-10-13 CODE ↗

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

Diagram Design adds architecture containment scopes and native Qoder plugin support

WAS THIS USEFUL? 👍 👎

keploy

Feedback ⓘ v3.6.215 NOTES ↗ RANK

Open-source platform for creating safe, isolated production sandboxes for API, integration, and E2E testing.

> Keploy extended its mock engine with macOS support for the agent and proxy modes and improved consistency of mock state across worker-mode test suites.

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

01 **Global mock serving in worker mode** IMPROVED 40

Keploy now lists mock calls that fall outside any individual test in worker mode and serves them globally, making shared mock state available across the full test suite.

— Describes behavior but no config or command to invoke it v3.6.215

02 **macOS support for mock flow** NEW 40

Keploy now runs the proven mock flow on macOS using both the agent and proxy modes.

— Names platform and modes but no setup steps v3.6.214

03 **Consistent HTTP mock store reads** IMPROVED 35

Keploy now answers a store's read with what the test wrote, keeping HTTP mock state consistent across record and replay.

— Explains mechanism but no named surface or command v3.6.214

WAS THIS USEFUL? 👍 👎

Alibaba OpenCodeReview

Feedback ⓘ v1.13.5 NOTES ↗

Secure, fast, efficient, battle-tested at Alibaba's scale. Hybrid architecture code review tool: deterministic pipelines + LLM Agent, precise line-level comments, built-in multi-language ruleset (NPE, thread-safety, XSS, SQL injection), OpenAI & Anthropic compatible.

└──▷ GET THIS VERSION ▾

Homebrew

```
$ brew install open-code-review
# already installed? upgrade to this version:
$ brew upgrade open-code-review
```

npm

```
$ npm install -g @alibaba-group/open-code-review@1.13.5
```

git

```
$ git clone --branch v1.13.5 https://github.com/alibaba/open-code-review.git
# already have the repo? check out this version:
$ git checkout v1.13.5
```

Homebrew lists this version · signed build provenance · checked 4 Nov Wrong command?

NEW SUPPORTED PLATFORM HAML, Slim, and ERB template file support

NEW SUPPORTED PLATFORM JSONC file support

NEW SUPPORTED PLATFORM EJS, Liquid, and Nunjucks template file support

NEW DETECTION Scala-specific review rules

WAS THIS USEFUL? 👍 👎

ogcode

Feedback ⓘ v0.58.2 NOTES ↗ RANK

An open-source, local-first AI teammate and agent operating environment for turning intent into completed work—with tools, memory, permissions, and human oversight.

> ogcode adds a workspace-scoped activity heatmap backed by a new per-day activity endpoint, while tightening validation across several tool schemas (browser, teammate, write, ask_user, glob, grep) and changing `bash` timeout semantics — several of these are breaking changes for existing callers.

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

01 **Stricter schema validation across tool calls** BREAKING

90

Several tool schemas now enforce validation at call time instead of leaving it to prose: `browser_navigate` requires a `url`, `browser_click` accepts a `ref` or a `selector`, and `browser_key` requires a key; the `teammate` schema now states per-operation requirements so a call carrying only a prompt is treated as a spawn; a `write` call whose `content` field is absent is refused outright while an explicit empty string `""` still writes an empty file; an `ask_user` batch that repeats a question id is now refused since answers are keyed by id; and a `glob` pattern the matcher cannot understand or a `grep` whose scan stopped at a long line now return errors instead of empty or partial results.

– Names every affected tool, field and behavior change verbatim v0.58.2

02 **`bash` zero/negative timeout means no deadline** BREAKING

80

A `bash` call with `timeout: 0` or a negative timeout now means no deadline rather than an instant expiry; an omitted timeout keeps the documented 120-second default.

– Exact flag, values, and default documented with clear before/after v0.58.2

03 **Home-screen activity heatmap and per-day endpoint** NEW

70

The home screen now shows a full-year activity heatmap, one cell per calendar day, shaded by activity level, backed by a new per-day activity endpoint that joins session records to the spend ledger. It is workspace-scoped and inclusive of task worktrees.

– Mechanism and scope given but no exact endpoint path or UI steps v0.58.2

04 **`path` field added to `query_sheet`** IMPROVED

60

The `query_sheet` tool now accepts the `path` field alongside its own `file` field, matching the spelling used by sibling index tools.

– Exact field name given but minimal explanation of mechanism v0.58.2

└──▷ **BREAKING ON UPGRADE**

! A `write` call that omits the `content` field is now refused rather than silently writing an empty file — callers that relied on the previous behavior must supply an explicit `content` field.

! A `bash` call with `timeout: 0` previously expired immediately; it now means no deadline, changing the behavior of any automation that used zero as an intentional instant-timeout sentinel.

! A `glob` pattern the matcher cannot understand and a `grep` whose scan stopped at a long line now return errors instead of empty or partial results, which may break callers that treated those responses as success.

WAS THIS USEFUL? 👍 👎

clay

Feedback ⓘ v5.6.0 NOTES ↗ RANK

Clay is a self-hosted workspace where people and coding agents work together across projects and sessions.

> Clay's latest release focuses on cosmetic workspace updates, adding new appearance options and chat surface tweaks.

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

01 **Nostalgic themes and curated wallpapers** NEW 35

Clay adds nostalgic color themes and curated wallpapers under appearance settings, giving users new visual customization options for the workspace.

– Names appearance settings area but no specific options listed v5.6.0

02 **Empty entry and translucent chat surfaces** NEW 20

Clay adds empty entry and translucent chat surfaces to the workspace.

– Bare description with no mechanism or location detail v5.6.0

WAS THIS USEFUL? 👍 👎

forgedock

Feedback ⓘ v1.44.0 NOTES ↗ RANK

ForgeDock — autonomous AI dev pipeline for Claude Code. GitHub as agent memory: issue in, PR out.

> ForgeDock improved scheduling visibility and dependency handling, with `sched show` now surfacing remote pool lanes and unresolved dependency commits triggering a hold instead of an outright launch refusal.

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

01 **Remote pool lanes shown in sched show** IMPROVED 55

The `sched show` command now includes remote (pool) lanes, so a pool-only batch no longer appears idle.

— Names exact command and behaviour change, no example given v1.44.0

02 **Admit-and-hold for unresolved dependency commits** IMPROVED 50

forgedock now admits and holds issues whose dependency commit is not yet on the base branch instead of refusing the launch outright.

— Explains before/after behaviour but no command or flag named v1.44.0

WAS THIS USEFUL? 👍 👎

octomind

Feedback ⓘ 0.69.0 NOTES ↗ RANK

Open-source AI coding agent and agent runtime: one binary, any model, MCP-native. Runs in terminal, CI, or as a daemon.

> octomind's 0.69.0 release migrates its ACP surface to agent protocol v1 and adds guidance for delegated agent runs.

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

01 **ACP migrated to agent protocol v1** IMPROVED 30

octomind's ACP surface now uses `agent protocol v1`.

— Names protocol version but no migration details or impact 0.69.0

02 **Guidance for delegated agent runs** NEW 10

octomind adds guidance for delegated agent runs.

— Only a bare mention with no mechanism or scope given 0.69.0

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ Service Accounts (2026-11-03)

BLOG ↗ RANK

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

> Amp introduces service accounts, giving teams workspace-owned identities with dedicated API keys so automated and CI-triggered threads belong to the organization rather than an individual user.

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

01 Service accounts with workspace-owned threads NEW

86

Amp now supports service accounts — workspace-owned accounts created by admins with their own API keys and third-party credentials — whose threads are visible and writable by all workspace members rather than tied to the user who created them. Use a service account's key by setting `AMP_API_KEY=<SERVICE_ACCOUNT_API_KEY>` together with `amp -ox`, so programmatically spawned threads (e.g. from CI on build failure) belong to the workspace instead of the individual who set up the pipeline.



Trigger a workspace-owned Amp thread from CI whenever tests fail, so the resulting thread belongs to the team rather than an individual.

```
$ run_tests
if tests_failed; then
  AMP_API_KEY=<SERVICE_ACCOUNT_API_KEY> \
  amp -ox "Tests failed. Is this a real regression or a flaky
test?"
fi
```

— Names exact env var and CLI flag with a runnable CI example

Service Accounts (2026-11-03)

WAS THIS USEFUL?  

Amazon Kiro

Feedback ⓘ

CLI: V3 by Default, Classic and Desktop App Retired (2026-11-03) BLOG ↗ RANK

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

> Kiro CLI ships V3 as the new default agent harness, retiring the Classic interface and desktop app while adding cloud session management, sandboxing, memory, and approval controls.

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

01 V3 becomes default harness, retiring Classic and desktop app

88

BREAKING

Kiro CLI V3 is now the default agent harness for interactive sessions, so the `--v3` flag is no longer needed. The Classic (V1) interactive interface no longer runs — scripts or aliases selecting it get a Continue with V3 prompt interactively, or the selection is silently ignored in non-interactive mode. The desktop app is retired; existing macOS desktop installs migrate to the standalone `kiro-cli` binary on their next update check. The `autocomplete`, `inline translate theme`, and `dashboard` commands are removed and must be deleted from scripts, along with inline shell autocomplete and inline suggestions, which have no standalone replacement. Headless scripts using `--trust-all-tools` now default to V3 instead of V2, and `kiro-cli doctor` no longer functions as before — it now redirects to `kiro-cli diagnostic` or `kiro-cli issue`. Users who need the old behaviour can run `kiro-cli --v2` to keep the V2 agent harness.

Fall back to the V2 agent harness when a custom agent or workflow is not yet compatible with V3.

```
$ kiro-cli --v2
```

— Names every removed command and migration flag with exact behaviour.

CLI: V3 by Default, Classic and Desktop…

02 Workflow plan preview before execution NEW

65

Turning on Workflows under `/settings > Features` shows a plan before any workflow starts executing.

Enable Workflows so Kiro shows a plan before executing, then navigate to the feature toggle.

📍 In the Kiro CLI, go to `/settings` › Features and enable Workflows.

– Exact navigation path and behaviour given. CLI: V3 by Default, Classic and Desktop…

03 **Sandboxed shell commands via /sandbox** NEW 60

The `/sandbox` subcommand isolates shell commands from the host environment.

– Names exact subcommand but no further mechanism given.

CLI: V3 by Default, Classic and Desktop…

04 **Resumable cloud sessions via /sessions** NEW 60

The `/sessions` subcommand lets you resume disconnected sessions, detaching from and reconnecting to cloud sessions.

– Exact subcommand named, behaviour described briefly.

CLI: V3 by Default, Classic and Desktop…

05 **'Approve for me' auto-approval mode** NEW 60

Enabling Approve for me lets Kiro auto-approve routine actions, block unsafe ones, and prompt on ambiguous ones; existing deny, ask, and allow rules still take precedence.

– Explains decision logic but not exact toggle location.

CLI: V3 by Default, Classic and Desktop…

THINNER COVERAGE BELOW

06 **Cross-session memory of repository context** NEW 50

Enabling the memory setting carries repository context forward across sessions.

– Describes effect but not the exact setting path or key.

CLI: V3 by Default, Classic and Desktop…

└──▷ **BREAKING ON UPGRADE**

! The Classic (V1) interactive interface no longer runs; scripts or aliases that select it will be offered Continue with V3 interactively, or the selection is silently ignored in non-interactive mode.

! The `autocomplete`, `inline translate theme`, and `dashboard` commands are removed and must be deleted from scripts.

! The desktop app is retired; existing macOS desktop installs will migrate to the standalone `kiro-cli` binary on their next update check.

! Inline shell autocomplete and inline suggestions are retired with no standalone replacement.

! `kiro-cli doctor` no longer functions as before; it now redirects to `kiro-cli diagnostic` or `kiro-cli issue`.

! Headless scripts using `--trust-all-tools` now default to V3 instead of V2.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

LangGenius Dify

Feedback ⓘ 1.19.0 NOTES ↗ RANK

Dify is an LLM application platform for building agentic workflows and RAG pipelines through a visual workspace, deployable to cloud or self-hosted.

> Dify 1.19.0 adds a Skill Hub CLI, nested workflow tracing for external observability tools, portable Workflow Tool packages for app export, and richer Agent file handling, alongside several deployment and config-precedence changes that require operator action.

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

01 Skill Hub management via difyctl CLI NEW

80

Workspace Skills can now be managed entirely from the command line with six new subcommands: `difyctl get skill`, `difyctl describe skill`, `difyctl get skill version`, `difyctl export skill`, `difyctl import skill --files <dir>`, and `difyctl publish skill`.

Create or replace a Skill's draft from local source files, then publish it as a new version.

```
$ difyctl import skill --files ./my-skill-dir && difyctl publish skill
```

— Lists all six commands with a runnable example, but no mechanism detail. 1.19.0

02 Rolling-upgrade tracing queue migration BREAKING

80

Tracing behaviour during rolling upgrades is now controlled via `OPS_TRACE_QUEUE=ops_trace_v2`; any custom `CELERY_QUEUES`, `CELERY_WORKER_QUEUES`, or `--queues` lists must include the new `ops_trace_v2` queue alongside the legacy `ops_trace` queue or new tracing tasks will be missed.

Override the tracing queue for the API and worker services so new trace tasks land on the v2 queue during a rolling upgrade.

```ini
OPS_TRACE_QUEUE=ops_trace_v2
```

— Names the exact env var and queue config with a before/after risk, plus example. 1.19.0

03 Nested workflow traces for external observability tools NEW

65

External observability tools now receive nested Workflow Tool execution steps, with child steps appearing beneath the Tool span that invoked them, including formatted prompts, responses, and token usage, down to the underlying LLM calls.

— Explains the trace structure and data shown, but no config name to act on. 1.19.0

04 **Portable Workflow Tool app packages (format v2)** BREAKING

65

App export now bundles the workflows behind Workflow Tools at their bound versions, including transitive dependencies, using a new package format v2; import reuses existing tools and recreates missing ones with remapped references. New packages always use format v2, so older destination importers may reject them — destination instances should be upgraded before importing packages produced by 1.19.0.

— Describes version/dependency behavior and migration risk, but no command or endpoint. 1.19.0

05 **Updated deployment build requirements** BREAKING

65

Docker Compose 2.35.0 or later is now the minimum supported version, since older versions no longer guarantee root `.env` overrides take precedence when Compose resolves service defaults. Separately, pnpm no longer provisions Node.js for source deployments — Node.js 24.x (24.22.1 or later) must be installed separately before running `pnpm install`.

— Names exact version thresholds for two unrelated build prerequisites. 1.19.0

06 **api-worker.env config precedence** IMPROVED

60

Shared API and worker configuration can now be customized through `docker/envs/core-services/api-worker.env`, which takes precedence over the legacy `docker/envs/core-services/shared.env` when both set conflicting values.

— Names both config files and the precedence rule but no usage example. 1.19.0

THINNER COVERAGE BELOW

07 **Agent file preview and batch upload improvements** NEW

55

Agent configuration files now support parallel batch upload with atomic binding, inline PDF preview, syntax-highlighted source file preview, and truncation indicators.

— Lists four UI capabilities without explaining mechanism or limits. 1.19.0

└──▷ BREAKING ON UPGRADE

! Docker Compose 2.35.0 or later is now the minimum supported version; older versions no longer guarantee that root `.env` overrides take precedence when Compose resolves service defaults.

! New app packages always use format v2; older destination importers may reject them — upgrade destination instances before importing packages produced by 1.19.0.

! New tracing tasks default to `OPS_TRACE_QUEUE=ops_trace_v2`; any custom `CELERY_QUEUES`, `CELERY_WORKER_QUEUES`, or `--queues` lists that do not include `ops_trace_v2` will miss new tracing work.

! pnpm no longer provisions Node.js for the workspace in source deployments — Node.js 24.x (24.22.1 or later) must be installed separately before running `pnpm install`.

WAS THIS USEFUL? 👍 👎

EXXETA exxperts

Feedback ⓘ v0.15.1 NOTES ↗ RANK

The exxperts framework runs local-first AI agents with approval-gated memory, model-provider integrations, MCP tools, and web search.

> exxperts replaced its terminal CLI with a browser-based app, added message editing and queuing, keyboard shortcuts, typeset maths/chart rendering, and a redesigned Attachments panel, while retiring dedicated chart, diagram and document specialists in favour of built-in generation.

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

01 **exxperts command now launches browser app** BREAKING

85

The `exxperts` command now opens the app in the browser instead of a terminal interface; web search, gateways and remote access are configured in the app's Settings. The `exxperts cli`, `exxperts remote`, and the search and gateway setup commands are removed.

— Exact removed commands named with clear migration path v0.15.0

02 **Edit sent and queued messages** NEW

85

Edit your most recent sent message with the pencil icon after the room has answered, or edit a queued message before the room takes it in; editing replaces the original message and its answer for both you and the model. A queued follow-up can also be sent immediately, stopping the current streaming answer, via Cmd+Enter (macOS) or Ctrl+Enter (Windows).

Send a follow-up message immediately and stop the current streaming answer at the words already on screen.

📍 While a room is answering, type your next message and press Cmd+Enter (macOS) or Ctrl+Enter (Windows) to stop the current answer and send at once.

— Exact shortcut and replacement mechanism given v0.15.0

03 **Keyboard shortcut reference** NEW

75

Open a keyboard shortcut reference with `Cmd+/` or `Ctrl+/` on Windows; Help now shows either How it works or the Shortcuts list.

View all available keyboard shortcuts without leaving the chat interface.

📍 Press Cmd+/ (macOS) or Ctrl+/ (Windows) while in the app to open the

Shortcuts list.

— Exact keyboard shortcut makes it immediately usable v0.15.0

04 **Chart, diagram and table rendering in answers** IMPROVED

70

Answers now render typeset maths and draw bar, line and pie charts, with Copy image, Code view, and a larger view available for diagrams. Charts, diagrams, and code now show tools in one slim strip with Copy always in the same position; tables and standalone formulas can now be copied, and Copy image keeps the full chart title and names.

— Named controls like Copy image but no API/config key v0.15.0 v0.15.1

05 **Attachments panel with gallery view** IMPROVED

60

The room Files list is renamed to Attachments and now shows pictures, text files, and a PDF's first page as cards, with a gallery view for a larger look and one menu to download, rename, or delete a file.

— UI panel described but no config or API surface v0.15.1

06 **Model and effort picker with context meter** NEW

60

Pick the model and set its effort level directly below the message box, which also shows each model's context capacity and the Ask first / Auto chip for commands.

— UI location given but no config key v0.15.0

07 **Resume answer after app quit** NEW

60

Quitting the desktop app mid-answer preserves the words written so far on Mac and Windows; typing continue resumes the answer after restart.

— Exact resume keyword makes it directly usable v0.15.0

THINNER COVERAGE BELOW

08 **Live thinking and step cards during streaming** NEW

50

Answers stream with live thinking and step cards: a Thinking timer line expands to show reasoning as it is written, then collapses to show total time (e.g. Thought for 7 s).

— Describes UI behaviour with no control exposed v0.15.0

09 **Folder write confirmation for unrequested files** IMPROVED

50

Rooms with a folder now ask before replacing a file that was not explicitly requested, or write under a new name; scheduled runs are instructed to make a reasonable choice and

finish rather than stop on an unanswerable question.

— Behavioral change with no configurable control named v0.15.1

10 **Chart and document specialists retired** DEPRECATED 50

Specialists for charts, diagrams, slide decks, and documents are switched off — a room now draws charts and diagrams directly in its own answers and writes files to a connected folder.

— Explains behaviour change, no toggle or config given v0.15.1

11 **Chat width setting** NEW 45

Set chat width to Standard or Wide in the new Appearance pane under Settings.

— Simple toggle fully specified, minimal mechanism v0.15.1

12 **Copy answers into email formatting** NEW 45

Copy answers into email keeping headings, lists, and tables without app colours — charts and diagrams are omitted from the copy.

— Behavior described, no exact UI path given v0.15.1

13 **GPT-6.1 Terra model added** NEW 40

GPT-6.1 Terra joins the model list on OpenAI, ChatGPT, Azure and GitHub Copilot.

— Names model and providers but nothing further v0.15.0

└──▷ **BREAKING ON UPGRADE**

! The room Files list is renamed to Attachments.

! The `exxperts cli`, `exxperts remote`, and the search and gateway setup commands are removed; the `exxperts` command now opens the browser app instead of a terminal interface, and web search, gateways and remote access must be configured in the app's Settings.

WAS THIS USEFUL?  

AutoGPT

Feedback ⓘ `autogpt-platform-beta-v0.11.0` NOTES ↗

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

AutoGPT Platform adds Parallel Decisions blocks, a rate-limited v1 external API, trusted-proxy client-IP resolution via `TRUSTED_PROXIES`, and a `MAX_REQUEST_BODY_MB` request-size limit.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch autogpt-platform-beta-v0.11.0
https://github.com/Significant-Gravitas/AutoGPT.git
# already have the repo? check out this version:
$ git checkout autogpt-platform-beta-v0.11.0
```

Adds Parallel Decisions blocks — web-grounded yes/no, choice, and score variants — for branching automation flows.

Rate limits the v1 external API.

Resolve real client IPs through proxy chains by listing trusted proxies in the new `TRUSTED_PROXIES` setting (IP addresses or CIDR ranges).

Enforce a request-body size limit on the platform API via `MAX_REQUEST_BODY_MB` (default 10 MiB), returning HTTP `413` on routes that are not upload or webhook-ingress endpoints.

RETIRED INTEGRATION Control payment enforcement on self-hosted installs with the `ENABLE_PLATFORM_PAYMENT` backend setting instead of a feature flag.

+3 more

NEW INTEGRATION Microsoft unified connection

UPDATED INTEGRATION AutoPilot traces moved from Langfuse to PostHog

NEW SUPPORTED PLATFORM Gemini 3.5 Flash, 3.5 Flash Lite, and 3.1 Flash Lite added to LLM catalog

└──▷ BREAKING ON UPGRADE

! `ENABLE_PLATFORM_PAYMENT` replaces the `enable-platform-payment` feature flag; `FORCE_FLAG_ENABLE_PLATFORM_PAYMENT` and `NEXT_PUBLIC_FORCE_FLAG_ENABLE_PLATFORM_PAYMENT` no longer do anything.

! The platform API now returns HTTP `413` for any request body exceeding `MAX_REQUEST_BODY_MB` (default 10 MiB) on non-upload, non-webhook routes.

- ! Deployments behind a proxy chain must list trusted proxies in `TRUSTED_PROXIES`; an invalid entry prevents the frontend from starting.
- ! LaunchDarkly support is removed; `LAUNCH_DARKLY_SDK_KEY`, `FEATURE_FLAG_BACKEND`, `NEXT_PUBLIC_LAUNCHDARKLY_ENABLED`, `NEXT_PUBLIC_LAUNCHDARKLY_CLIENT_ID`, and `NEXT_PUBLIC_FEATURE_FLAG_BACKEND` are ignored.
- ! AutoPilot conversation traces no longer go to Langfuse; they now go to PostHog.

WAS THIS USEFUL? 👍 👎

cotal

Feedback ⓘ v0.141.0 NOTES ↗ RANK

The open standard for agent coordination

> Cotal shipped inline persona definition and capability-bounded helper spawning, configurable history limits spanning DMs and delivery streams, per-agent dashboard pages with live tool-call visualization, and a new desktop daemon app with system tray and device-approval onboarding.

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

01 Configurable history limits for DMs and streams

BREAKING

100

`cotal history limit` now controls inbox and stored-copy limits with `--inbox-max-age` and `--inbox-per-agent`, and DM limits with `--dm-max-age` and `--dm-per-sender`; passing `none` to any argument removes that limit, and running the command with no flags prints current limits for all four streams. `cotal history limit --json` now returns one entry per stream under `result.streams` instead of a single result, and the delivery daemon stamps each stored copy with a `Cotal-Delivery-Limit` header so the stored-copies notice can compare against the original limit, surfaced as the stamped `limit` field on the `stored-copies-full` event. In `@cotal-ai/core`, history-limit control is generalized to any stream via `setHistoryLimit`, `describeHistoryLimit`, `historyLimitProblem`, `historyLimitOf`, and `historyLimitRestoreStaging`, and new types `HistoryLimit`, `HistoryLimitChange`, `HistoryLimitUpdateUnknown`, `HistoryLimitChanged`, replacing the chat-only `ChatHistoryLimit` (renamed, with `perSender` becoming `perSubject`) and removing `setChatHistoryLimit`, `describeChatHistory`, `chatHistoryLimitProblem`, `chatHistoryLimitOf`, `chatHistoryRestoreStaging`, `ChatHistoryChange`, `ChatHistoryUpdateUnknown`, `ChatHistoryChanged`, `isAcceptedChatHistory`, `CHAT_HISTORY_MIN_AGE_MS`, and `CHAT_HISTORY_MAX_AGE_MS`; an age value can be checked in isolation with `historyAgeProblem` (`0` signals below the minimum floor) against bounds `historyLimitMinAgeMs` and `HISTORY_MAX_AGE_MS`.

Cap how many stored copies the delivery daemon keeps per agent and how long DMs are retained, in one command.

```
$ cotal history limit --inbox-per-agent 500 --inbox-max-age 30d --dm-max-age 90d
```

Remove all age and count limits on DMs so no DM history is ever pruned.

```
$ cotal history limit --dm-max-age none --dm-per-sender none
```

Inspect current limits across all four streams in machine-readable form for a monitoring script.

```
$ cotal history limit --json
```

— Enumerates every flag, function and type renamed or added `v0.140.0`

02 **Inline persona definition for spawned helpers** NEW

85

`cotal_spawn` now accepts a `persona` record (prompt, role, model, variant, description, channels, capabilities) alongside `name`, letting a helper run from a private snapshot without a catalog entry; setting `persona.save: true` persists the copy to the catalog, owned by the spawner. The same inline definition works in cotal-lang via spawn({ name, prompt, ... }, opts), with the run journal recording the definition and its digest so a resume never resubmits it. A saved persona can be removed via `cotal_persona`, which now also accepts a `capabilities` field.

Spawn a scoped helper with a fully defined persona inline, without pre-registering it in the catalog.

```
$ cotal_spawn({ name: "analyst", persona: { prompt: "You are a triage analyst.", role: "analyst", model: "gpt-4o", channels: ["alerts"], capabilities: ["read:alerts"], save: false } })
```

— Names exact fields and commands, with a runnable example `v0.141.0`

03 **Capability ceiling enforcement for spawned helpers** NEW

83

`cotal run start --local` now bounds helper channel access to within the `--admit-read` and `--admit-publish` ceiling granted to the starter, preventing a helper from exceeding the starter's access. A static spawner that attempts to hand a helper channels, capabilities, or a role it does not itself hold is now refused, and an admission carrying `spawnExempt` or `delegation` is refused by an older reader.

Run a local workflow whose helpers are ceiling-bounded to specific read and publish channels, preventing privilege escalation.

```
$ cotal run start --local --admit-read alerts,reports --admit-publish findings my-workflow.cotal
```

— Names exact flags and refusal conditions, with example v0.141.0

04 **Indefinite lifecycle credentials for foreground seats** NEW

82

`cotal spawn` or `cotal join` on a static mesh now runs under its own lifecycle credential: the credential has no expiry unless `--expires-in` or `--expires-at` is passed, is never renewed, and is retired automatically when the seat leaves cleanly. An older issuing host refuses the new enrollment field this introduces, and an enrollment retry is refused when its read set does not match its grant's read set, affecting grants an older host wrote for a helper named only with a subscribe set.

Spawn a foreground seat on a static mesh with an explicit expiry rather than the new indefinite-lifetime default.

```
$ cotal spawn --expires-in 8h
```

— Names flags, default behaviour and compatibility edge cases v0.141.0

05 **Per-agent dashboard pages with live work visualization** NEW

80

The `cotal web` dashboard gives each agent a dedicated page at `#/agent/<id>`, reached from the Agent tab's peek, its 'Open <name>'s page' button, or the View work action on any live message from that agent. The page's Work tab groups tool calls by turn and shows each as running, done, or failed with elapsed time, driven by a new Claude Code connector event — a `CUSTOM` event named `cotal.toolResult` published at each tool result, carrying `{ toolCallId, isError }` and a timestamp (no tool input/output included) — which `cotal console` also draws as a `done` / `failed` marker beneath each call. The agent card exposes `meta['ai.cotal.events']` set to `on` or `off` to distinguish an agent that publishes no events from one that hasn't started a turn, agent prose gets syntax-highlighted code blocks, and the dashboard keeps the newest 500 direct messages, lists the ten most recent conversations in the sidebar, and shows a working agent as Working without a progress unknown suffix. The new components are built on shadcn/ui and Tailwind, scoped so existing styles and installer size are unaffected.

— Names exact UI paths and event payload but only a navigation path v0.139.0

06 **Standalone attach and chat workspace libraries** NEW

78

Terminal-attach logic is now available as a standalone library via `@cotal-ai/workspace/attach`, exporting `attachClient`, `meshSessionTransport`, and `attachSeat`, and chat-session logic via `@cotal-ai/workspace/chat`, exporting `openChatSession` plus the open, static, and user-mode adapters — both letting callers embed these workflows without importing CLI code.

Open a chat session from library code without importing CLI code — openChatSession picks the right adapter from the target's recorded mode.

```javascript
import { openChatSession } from '@cotal-ai/workspace/chat';
const session = await openChatSession(controlClient, targetId);
```

— Names both subpaths and every export, with runnable example `v0.138.0`

07 **Automatic default persona seeding for cotal spawn** IMPROVED

75

Running `cotal spawn` in a Cloud space with no default persona now writes the shipped default persona to the catalog and launches it, instead of failing with `TARGET_NOT_FOUND`; pass `--dry-run` to preview the launch without writing anything.

Verify what cotal spawn would do in a fresh catalog without committing any writes.

```
$ cotal spawn --dry-run
```

— Names the flag, prior error, and new default behaviour `v0.136.6`

08 **Per-publisher channel history filtering** IMPROVED 72

In `@cotal-ai/core`, `channelHistory` now accepts a `publisher` option to read a channel's history from a single principal's subject only, and the history `limit` is capped at 2,000.

Read only one agent's history from a channel when auditing its activity, using the new `publisher` filter in `channelHistory`.

```javascript
channelHistory(channel, { publisher: 'reviewer_2', limit: 200 })
```

— Named function, option and numeric cap with runnable code `v0.139.0`

09 **Structured result from describeSendChannel** BREAKING

65

`MeshAgent.describeSendChannel` now returns `{ known, hints }` instead of a plain sentence, giving callers structured access to channel-resolution results; callers that read the return value as a string will break.

— Exact before/after return shape named, no usage example given `v0.138.0`

10 **Broker allocation profiling for smoke runs** NEW

60

The `--broker-prof` flag writes the broker allocation profile during a smoke run, for inspecting resource distribution.

Capture the broker allocation profile while running a smoke test to inspect resource distribution.

```
$ cotal smoke --broker-prof
```

— Named flag with runnable command, but thin on mechanism `v0.136.3`

THINNER COVERAGE BELOW

11 **Expanded usage guidance for coordination tools** IMPROVED

53

`cotal_send`, `cotal_dm`, `cotal_anycast`, `cotal_spawn`, `cotal_persona`, `cotal_despawn`, `cotal_join`, `cotal_leave`, `cotal_channels`, `cotal_channel_info`, and `cotal_run` now describe their purpose, call out when a neighbouring tool fits better, and include one usage example each.

— Names every tool but describes only documentation, no mechanism `v0.138.0`

12 **Canonical age display across the dashboard** IMPROVED

52

Dashboard roster, waiting cards, and agent details now show reported conditions, status, activity, and work ages using canonical age functions via a new browser-safe `presence-ages` subpath, so a 90-second heartbeat displays as `1m` rather than rounding to `2m`.

— Concrete before/after example but only UI-facing, no entry point `v0.138.0`

13 **Desktop daemon app with system tray** NEW

43

cotal now includes a desktop daemon app with a system tray, device-approval onboarding, daemon-owned space discovery, and connection controls, with the stock CLI bundled inside the app.

— Names feature set but no commands, flags or screenshots given `v0.137.0`

└──▷ BREAKING ON UPGRADE

- ! A static spawner that attempts to hand helpers channels, capabilities, or a role it does not itself hold is now refused.
- ! An older issuing host refuses the new enrollment field introduced in this release.
- ! An older manager refuses a run's spawn, which now names its run.
- ! An older reader refuses an admission carrying `spawnExempt` or `delegation`.
- ! An enrollment retry is refused when its read set does not match its grant's read set — this affects grants an older host wrote for a helper that named only a subscribe set.
- ! In `@cotal-ai/core`, `ChatHistoryLimit` is renamed to `HistoryLimit` and its `perSender` field is renamed to `perSubject`.
- ! In `@cotal-ai/core`, `setChatHistoryLimit`, `describeChatHistory`, `chatHistoryLimitProblem`, `chatHistoryLimitOf`, and `chatHistoryRestoreStaging` are removed; use `setHistoryLimit`, `describeHistoryLimit`, `historyLimitProblem`, `historyLimitOf`, and `historyLimitRestoreStaging` instead.
- ! In `@cotal-ai/core`, `ChatHistoryChange`, `ChatHistoryUpdateUnknown`, and `ChatHistoryChanged` are removed; use `HistoryLimitChange`, `HistoryLimitUpdateUnknown`, and `HistoryLimitChanged` instead.
- ! In `@cotal-ai/core`, `isAcceptedChatHistory`, `CHAT_HISTORY_MIN_AGE_MS`, and `CHAT_HISTORY_MAX_AGE_MS` are removed; use `historyAgeProblem`, `historyLimitMinAgeMs`, and `HISTORY_MAX_AGE_MS` instead.
- ! `cotal history limit --json` result shape changes: the previous single-result format is replaced by one entry per stream under `result.streams`.
- ! A Claude Code connector downgraded across a write-ahead log that already holds a `cotal.toolResult` marker refuses the event name and halts.
- ! A frozen write-ahead log frame from an older connector version that still carries `rawEvent` halts recovery on every restart.
- ! `MeshAgent.describeSendChannel` now returns `{ known, hints }` instead of a sentence string — callers that read the return value as a string will break.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.9.10 NOTES ↗ RANK

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

> HyperFrames added terminal-based billing with price display and a renamed `billing add-credits` command, time-shifted frame comparison via `diff --offset`, plan-member style template pulling, forced per-render screenshot capture in Producer, HeyGen-account audio stem reading without a Gemini key, and reverse-run recording that carries sessions into cloud render.

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

01 Terminal billing with add-credits command BREAKING

80

`hyperframes billing` now displays the price and confirms payment on Enter from the terminal. The purchase command `billing top-up` has been renamed to `billing add-credits` with no alias, so any script or workflow calling `billing top-up` will break.

Top up workspace credits without leaving the terminal — the new command shows the price before charging.

```
$ hyperframes billing add-credits
```

— Names exact renamed command and behaviour, but no deeper mechanism given v0.9.10

02 Time-shifted frame comparison with diff --offset NEW

75

The `diff --offset` flag compares the original frame at time t against yours at t + offset, letting you spot drift between a reference cut and an edit that are not aligned to the same start time.

Spot drift between a reference cut and your edit when the two clips are not aligned to the same start time.

```
$ hyperframes diff --offset 2.5 reference.mp4 mine.mp4
```

— Mechanism and runnable example given, limited further detail v0.9.9

03 Audio stem reading via HeyGen account NEW

70

`hyperframes audio --read` describes audio stems in a project using your HeyGen account credentials, removing the need for a Gemini key.

Inspect audio stems in a project using your HeyGen account credentials instead of a third-party AI key.

```
$ hyperframes audio --read
```

— Exact command and a concrete before/after on key requirements v0.9.6

04 **Reverse run recording with cloud render session carry** NEW

70

`reverse start` records a reverse run and carries its session through cloud render; the run halts automatically if the account is on the Free tier.

Start a reverse run so the session persists into cloud render, or halt automatically if the account is on the Free tier.

```
$ reverse start
```

— Exact command with a specific tier-based behaviour noted v0.9.6

05 **Style template pulling for plan members** NEW

60

As a HyperFrames plan member, `hyperframes templates pull` downloads your plan's style templates locally so they can be applied via the style-sampling skill.

Download your plan's style templates locally so the style-sampling skill can apply them in a new film.

```
$ hyperframes templates pull
```

— Clear command but plan requirement and skill mechanism only lightly described v0.9.9

THINNER COVERAGE BELOW

06 **Per-render forced screenshot capture in Producer** NEW

45

HyperFrames' Producer now lets a render request force screenshot capture for that specific render, as an alternative to BeginFrame capture.

— Describes the change but no command or config surface to act on v0.9.8

07 **Blurred loading frame in player** IMPROVED

30

The player now shows a blurred frame and mark while assets load, matching Desktop behaviour.

— Describes the visual change only, no interaction surface `v0.9.6`

08 **Early-error kind and origin recording in Studio Server previews**

`IMPROVED` 25

HyperFrames' Studio Server previews now record the kind and origin of each early error.

— Bare statement with no mechanism or usage path `v0.9.8`

└──▷ **BREAKING ON UPGRADE**

! `billing top-up` is renamed to `billing add-credits` with no alias — any script or workflow calling `billing top-up` will break.

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ `langchain-typesafe==0.0.1` NOTES ↗ RANK

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

> LangChain introduced a new langchain-typesafe library adding classifier and middleware classes for safer, structured LLM routing.

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

01 **New langchain-typesafe library for LLM routing** NEW

62

LangChain shipped the new `langchain-typesafe` package with three classes: `TypeSafeClassifier`, which classifies LLM inputs via invocation-scoped classifier questions for structured routing decisions; the experimental `ModelRouterMiddleware`, which routes requests across models at runtime; and the experimental `AutoModeMiddleware`, which automatically selects a mode or model based on request context.

— Names three classes but no usage examples or mechanism detail.

`langchain-typesafe==0.0.1`

WAS THIS USEFUL? 👍 👎

sympozium

Feedback ⓘ v0.13.0 NOTES ↗ RANK

sympozium is a coordination layer for multi-agent AI systems

> sympozium v0.13.0 introduces a pluggable ExecutionBackend seam with a hardware-isolated Celln microVM backend (now the default), adds xAI (Grok) as a model provider, and ships several breaking changes to model-key and run-level configuration.

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

01 ExecutionBackend seam with Celln microVM backend NEW

95

A new ExecutionBackend seam lets tasks run via `backend: celln` in an `AgentRun` spec, executing in hardware-isolated microVM cells. The Celln fleet can be enabled at install time with `sympozium install --celln-fleet`, and the Celln fleet opt-in together with the Job backend are now the default, so existing installs that relied on the previous default backend behavior will need to review their configuration.

Deploy sympozium with the Celln fleet enabled so agents run in hardware-isolated microVM cells on KVM-capable nodes.

```
$ sympozium install --celln-fleet
```

— Names config key, spec field, and exact install command v0.13.0

02 Model key origin annotation required for custom baseURL

90

BREAKING

A model key used with a custom `baseURL` (self-hosted gateway, Azure, or local model server) must now be pinned via `kubectl annotate secret <name> sympozium.ai/model-key-origin=<origin>`.

— Exact command and annotation key given for migration v0.13.0

03 Run-level configuration moved to Agent level BREAKING

70

Run-level proxy/`SSL_CERT` env vars, Secret/PVC/CSI volumes, Secret-reading hooks, and `lifecycle.rbac` must now be set on the Agent instead of the run — they are no longer permitted at the run level.

— Lists all affected surfaces but no exact migration syntax v0.13.0

THINNER COVERAGE BELOW

04 **xAI (Grok) model provider support** NEW 45

sympozium now supports xAI (Grok) as a model provider alongside existing providers, and a related fix ensures the system no longer drops provider-named model keys.

— Names the provider but no config/command shown v0.13.0

└──▷ **BREAKING ON UPGRADE**

! A model key used with a custom `baseURL` (self-hosted gateway, Azure, or local model server) must now be pinned via `kubectl annotate secret <name> sympozium.ai/model-key-origin=<origin>`.

! Run-level proxy/`SSL_CERT` env, Secret/PVC/CSI volumes, Secret-reading hooks, and `lifecycle.rbac` must move onto the Agent (they are no longer permitted at the run level).

! The Celln fleet opt-in (`--celln-fleet`) and the Job backend are now the default — existing installs that relied on the previous default backend behavior will need to review their configuration.

WAS THIS USEFUL? 👍 👎

Brig

Feedback ⓘ v0.5.1 NOTES ↗ RANK

Brig runs coding agents in a microVM sandbox, made up of the brig CLI and the brigd session daemon.

> Brig v0.5.1 adds per-route TLS mutual authentication with certificate expiry warnings, AWS credential vending, a project path environment variable for sandboxes, and a Linux prerequisite checker.

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

01 **Per-route TLS mutual authentication** NEW 80

Control TLS mutual authentication per route with the new `inject.tls` field in the route schema. `brig wrap` holds certificate secrets on the host and warns when a client certificate will expire within 3 days.

– Names config field and command with concrete expiry threshold v0.5.1

02 **Secret type checking for TLS certificates** NEW 65

Check a secret's certificate or CA bundle type with `--type` on `brig secret`, with `tlspem` validation for certificate secrets.

– Names flag and validation type, lacks broader mechanism v0.5.1

03 **Linux host prerequisite checks in brig doctor** NEW 60

Run `brig doctor` on Linux to check host prerequisites a Linux run needs, including AppArmor profiles, CNI plugins, and rootless-run device access.

– Names command and three checked prerequisites v0.5.1

THINNER COVERAGE BELOW

04 **BRIG_PROJECT environment variable** NEW 45

Set the `BRIG_PROJECT` environment variable inside the sandbox to the guest path of the mounted project.

– Names env var but no further mechanism given v0.5.1

05 **AWS credential vending via aws CLI** NEW 30

Brig adds AWS credential vending via the `aws` CLI.

– Only mentioned in summary with no mechanism detail v0.5.1

WAS THIS USEFUL? 👍 👎

LangChain LangGraph

Feedback ⓘ sdk==0.4.12 NOTES ↗

RANK

Low-level orchestration framework for building, managing, and deploying long-running, stateful AI agents.

> LangGraph's SDK release simplifies run priority configuration and improves stream reliability for long-running runs.

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

01 Removed fairness_weight from PrioritySettings

BREAKING

50

The `fairness_weight` field has been removed from `PrioritySettings` to simplify run priority configuration. Any code that sets this field will break on upgrade.

— Names exact field and config object but no replacement guidance sdk==0.4.12

02 Automatic rejoining of run streams

IMPROVED

25

The SDK now automatically rejoins run streams that close before the run ends.

— States behavior change but no mechanism or API detail sdk==0.4.12

└──▷ **BREAKING ON UPGRADE**

! The `fairness_weight` field has been removed from `PrioritySettings`; any code that sets this field will break on upgrade.

WAS THIS USEFUL? 👍 👎

openexecutive

Feedback ⓘ v0.6.9 NOTES ↗ RANK

AI-powered virtual executive team — a single coherent executive persona backed by 8 specialist agents (FastAPI + Next.js).

> OpenExecutive's v0.6.9 release adds safeguards for externally-shared Drive folders, richer HTML tables in Act as me email drafts, more transparent progress reporting for long-running chat turns, and new search, sort, and change-tracking tools on the People page.

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

01 **External-share confirmation for Drive folder saves** NEW 52

OpenExecutive now shows a confirmation card and asks for confirmation before saving into a Drive folder that is shared outside the team, reducing accidental external data exposure.

— Describes trigger and UI mechanism but no config detail v0.6.9

02 **Live plan and progress view for long-running chat turns** IMPROVED 45

Long-running chat turns now show their plan up front before execution begins, and display what they are actively doing and have found as they run, rather than a static step list.

— Explains behaviour change but no concrete UI path or command v0.6.9

03 **HTML tables in Act as me email drafts** IMPROVED 38

Act as me email drafts now send tables as real HTML tables instead of plain text.

— Clear before/after but no further mechanism given v0.6.9

04 **Search, sort and Changes tab on People page** NEW 34

The People page gains search, sort, and a new Changes tab, per the release summary.

— Only named in summary, no elaboration on mechanism v0.6.9

WAS THIS USEFUL? 👍 👎

agentlas-os

Feedback ⓘ os-v1.2.80 NOTES ↗ RANK

Agent OS: keep specialist agents in a hub, spin up a temporary orchestrator per task. Local-first, works with any model.

> Agentlas OS introduces a machine-local memory system with new MCP tools and CLI commands, then quickly refines how that memory attributes work to the runtime that performed it.

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

01 Machine-local memory tools via MCP and CLI NEW

95

Models can search and manage machine-local memory via five new MCP tools: `memory.search`, `memory.timeline`, `memory.get`, `memory.crosscheck`, and `memory.record` — search works by meaning and keywords in any language, shows dates and match strength, and never returns a forgotten or replaced memory. New CLI commands `agentlas-os memory index`, `memory search`, and `memory timeline` let practitioners query and browse the memory store from a terminal. Memory is now on by default and belongs to the local machine rather than a signed-in account, with Agentlas Desktop and the standalone engine sharing the same store; it is no longer injected into every prompt — sessions start with a short note about the memory tools, and the model records its turn with `memory.record` instead of printing a memory ticket.

Browse recent memory entries from the terminal to audit what the model has recorded across sessions.

```
$ agentlas-os memory timeline
```

Search machine-local memory by keyword or meaning from the terminal to retrieve past context before starting a new session.

```
$ agentlas-os memory search
```

— Names every MCP tool and CLI command with runnable examples. os-v1.2.79

THINNER COVERAGE BELOW

02 Memory turns attributed to runtime, not agent IMPROVED

55

Memory turns are now recorded under the runtime that performed the work — `host:claude`, `host:codex`, and so on — instead of the personal agent, for both the

session hook and `memory.record`.

— Clear before/after but no user-facing action to take. os-v1.2.80

WAS THIS USEFUL? 👍 👎

Feynman Feedback ⓘ v0.5.62 NOTES ↗ RANK

Feynman is a personal AI research assistant that searches papers and the web, builds literature reviews, and produces cited research reports.

> Feynman added support for a new model provider (GPT-6.1 Terra) and improved tool argument handling to accept single values where lists are expected.

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

01 GPT-6.1 Terra model support NEW

65

Feynman now supports GPT-6.1 Terra as a model provider through OpenAI, Azure OpenAI, and OpenAI Codex. Feynman recommends `openai/gpt-6.1-terra` as the default model when only an OpenAI key is present.

— Names exact model id and providers but no setup steps v0.5.62

THINNER COVERAGE BELOW

02 Single-value coercion for list tool arguments IMPROVED

50

Tool arguments that expect a list, such as a paper section, now accept a single value and are automatically wrapped into a list. Validation errors for missing values are rendered as `- paper: missing`.

— Describes behavior and error format but no usage example v0.5.62

WAS THIS USEFUL? 👍 👎

yoagent

Feedback ⓘ v0.26.8 NOTES ↗ RANK

yoagent is a Rust agent loop that streams from 7 LLM protocols and runs tools until done, from a laptop to a Cloudflare Worker.

> yoagent shipped environment isolation controls for MCP stdio servers and added AWS Bedrock and OpenRouter gateway routing alongside new Claude and Gemini model presets.

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

01 Environment isolation for MCP stdio servers NEW 95

Callers can now strip or allowlist environment variables before launching an MCP stdio server: StdioOptions::env_remove(name) and StdioOptions::env_remove_all(names) unset variables entirely (not just empty them), StdioOptions::env_clear_except(names) passes only an allowlist and blocks everything else, and StdioOptions::new().env_remove_all(provider::api_key_variables()) strips every provider API key yoagent knows about. These options can be applied directly on the Agent path via Agent::with_mcp_server_stdio_with(command, args, env, options).

Prevent host API keys from leaking into an MCP stdio server by stripping every provider key yoagent knows about before launch.

rust

```
let options =
StdioOptions::new().env_remove_all(provider::api_key_variables())
;
let agent = Agent::with_mcp_server_stdio_with(command, args, env,
options);
```

Lock an MCP stdio server to only the variables it needs, blocking everything else from the host environment.

rust

```
let options =
StdioOptions::new().env_clear_except(["MCP_SERVER_TOKEN",
"HOME"]);
let agent = Agent::with_mcp_server_stdio_with(command, args, env,
options);
```

— Names exact methods with runnable code examples and clear mechanism. v0.26.8

02 AWS Bedrock and OpenRouter model gateways plus new presets

NEW

95

Requests can now be routed through AWS Bedrock — including cross-Region inference profiles like `us.anthropic.claude-sonnet-4-6` and `global.…` — via ModelConfig::preset("bedrock", id), which maps to Claude preset limits and bills Bedrock's own rates with a 10% surcharge on regional profiles. They can also be routed through OpenRouter via ModelConfig::preset("openrouter", "vendor/model"), accepting ids like `anthropic/claude-sonnet-4.6`, `openai/…`, `google/…`, `x-ai/…`, `z-ai/…`, and `qwen/…`, each mapped to the corresponding vendor preset's limits. New built-in presets ModelConfig::claude_sonnet_4_6() (1M context, adaptive thinking), ModelConfig::gemini_3_6_flash(), and ModelConfig::gemini_3_1_pro_preview() were added, with pricing read from vendor pages and activated via prices::enable_bundled().

Run an agent through AWS Bedrock using a cross-Region inference profile instead of a direct provider key.

```rust
let mut agent = Agent::from_config(ModelConfig::preset("bedrock",
"us.anthropic.claude-sonnet-4-6"))
    .with_system_prompt("You are a coding assistant.")
    .with_tools(tools::default_tools());
```

Route an agent through OpenRouter to access a Qwen model under a single gateway key.

```rust
let mut agent =
Agent::from_config(ModelConfig::preset("openrouter",
"qwen/qwen3.6-plus"))
    .with_system_prompt("You are a helpful assistant.");
```

Use the new Claude Sonnet 4.6 preset (1M context) for a long-context analysis task.

```rust
let mut agent =
Agent::from_config(ModelConfig::claude_sonnet_4_6())
    .with_system_prompt("You are a coding assistant.")
    .with_tools(tools::default_tools());
```

— Names exact preset calls, model ids, and billing detail with runnable examples. v0.26.7

WAS THIS USEFUL? 👍 👎

Cloudflare Agents

Feedback ⓘ agents@0.33.0 NOTES ↗ RANK

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

> Cloudflare Agents' 0.33.0 release adds Pi 1.5 support, replacing the session event stream with a new conversation-view API (session.watch(), session.continue(), session.entries()), alongside large storage and scheduler performance gains, new browser-tool session sharing, and several breaking API changes including an irreversible upgrade path.

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

01 Session lifecycle API: watch, continue, entries

BREAKING

90

session.continue() resumes a stalled agent run without sending a new message, useful after a model error or turn-limit hit. session.watch() replaces the removed session.events(), giving Pi's conversation view — live state, the streaming answer, running tools, and the reason a run ended without an answer — and session.entries() pages older transcript entries after switching to session.watch(). The old retry() method and `parked` pending status are removed; call session.continue() or resubmit instead. The `onReport` callback passed to `Harness.open` now receives one report tagged by `kind`, with `session_failed` renamed `storage_failed` and `task_parked` renamed `task_interrupted`. In pi-durable 1.5, watchEvents(), tagged watch calls/ends, and `ConversationView.entries` as a full transcript are removed; `ConversationView.entries` is now a window over the transcript, and an idle `pi.live` now keeps `runFailure` after a failed run — check `run === undefined` instead of comparing it to `{}`.

Retry an agent run that hit its turn limit without sending a new user message.

typescript

```typescript
const watch = await session.watch();
render(watch.value);
watch.start(async (call) => {
  if (call.kind === 'value') {
    render(call.value);
    if (call.value.runFailure?.reason === 'turn_limit') {
      await session.continue();
    }
  }
});
```

— Full before/after and many named APIs, with runnable example `agents@0.33.0`

02 **Browser session sharing in Pi browser tool** NEW 80

Code outside `browserTool` can now share the same `browser` object via Browser.functionDocs(), `BrowserSession.functions`, browser.session({ lock: true }), session.lock(), and conversationScope() from `agents/browser`. The Pi `browserTool` also now gives code-mode scripts the run as an object with screenshots as image blocks for image().

Let code outside browserTool share the same browser object by using conversationScope and session locks.

typescript

```
import { conversationScope, BrowserSession } from
'agents/browser';

const scope = conversationScope();
const bs = new BrowserSession(scope);
const locked = await browser.session({ lock: true });
await locked.lock();
// your code now holds the same browser object browserTool uses
```

— Several named functions with a runnable sharing example `agents@0.33.0`

03 **Pi 1.5 storage and scheduler performance overhaul** IMPROVED 75

Storage writes are now batched at most once per second, cutting SQLite transactions about 27x for long answers and 5.7x for turns with parallel tools, and reducing CPU by about a quarter. The scheduler shares one storage call for commits that arrive together and makes no storage reads on a normal run, raising throughput from 88 to 459 agent runs per second in a harness with 2,048 sessions. Streamed text is now appended instead of copied on every update, so large answers — a 500 KB tool call or a 140 KB answer followed by a 140 KB tool call — no longer spike memory to 100 MB.

— Rich numbers but automatic, no reader action needed `agents@0.33.0`

04 **Pi tool schemas via @earendil-works/pi-schema** BREAKING 73

Pi tool schemas are now built with `@earendil-works/pi-schema` using its obj() and str() helpers, since `pi-ai` no longer exports TypeBox's `Type`; import `Type` directly from `typebox` if still needed.

— Names package, helpers, and exact migration path `agents@0.33.0`

THINNER COVERAGE BELOW

05 **Reject input on idle session flag** NEW 55

Passing `whenIdle: "reject"` to submit() refuses new input instead of queuing it when the session is idle.

— Exact flag named but no example shown `agents@0.33.0`

06 **Pi 1.5 upgrade is irreversible** BREAKING 40

Once a conversation runs on Pi 1.5, Pi 1.4 cannot read it, so deployments cannot roll back past this release.

— Clear migration warning but no named surface `agents@0.33.0`

└──▷ **BREAKING ON UPGRADE**

! session.events() is removed; replace it with session.watch() and render the view each time it changes.

! retry() and the `parked` pending status are removed; call session.continue() or resubmit to run again after a run ends without an answer.

! pi-ai no longer exports TypeBox's `Type`; import tool-schema builders from `@earendil-works/pi-schema` instead, or import `Type` directly from `typebox`.

! The `onReport` callback passed to `Harness.open` now receives one report tagged by `kind` with the error on it; `session_failed` is renamed `storage_failed` and `task_parked` is renamed `task_interrupted`.

! watchEvents(), tagged watch calls and ends, and `ConversationView.entries` as a full transcript are removed in pi-durable 1.5; `ConversationView.entries` is now a window over the transcript.

! An idle `pi.live` now keeps `runFailure` after a failed run; check `run === undefined` instead of comparing it with `{}`.

! Once a conversation runs on Pi 1.5, Pi 1.4 cannot read it — deployments cannot roll back past this release.

WAS THIS USEFUL?  

pi-fabric

Feedback ⓘ v0.132.1 NOTES ↗ RANK

A programmable tool and agent runtime for Pi

> pi-fabric v0.132.1 introduces real-time observation APIs for background tasks and agent runs, plus a new command to demote foreground workflows to the background, alongside smaller UI and internal replay-engine refinements.

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

01 **Observation APIs for tasks and agents** NEW 75

New `tasks.observe` and `agents.observe` APIs let you watch running background tasks and local agent runs without a separate monitor tool. Observations support bounded event replay, explicit UI/wake delivery, and independent cancellation, so stopping or expiring an observation leaves the underlying target running.

— Named APIs and behaviour given, but no code sample v0.132.1

02 **Move foreground workflows to background** NEW 70

The `/fabric background` command moves an entire foreground workflow out of Main, including pending shell and agent waits. Running tool cards now also show the connected Ctrl+B background shortcut for quick access.

— Exact command and shortcut given, mechanism partly explained v0.132.1

THINNER COVERAGE BELOW

03 **Version-derived replay-engine identifiers** IMPROVED 40

Durable replay-engine identifiers now derive automatically from the package version, preserving replay fencing across releases without a manually maintained version literal.

— Internal mechanism explained but not user-actionable v0.132.1

04 **Task UI observation grouping** IMPROVED 35

The task UI now groups observations beneath their targets and distinguishes work counts from observation counts.

— Describes UI change but no navigation path or specifics v0.132.1

WAS THIS USEFUL? 👍 👎

truefoundry TrueForge

Feedback ⓘ

`@truefoundry/trueforge-ui@0.183.0` NOTES ↗ RANK

The open-source agent harness - the runtime layer that turns an LLM into a working agent.

> TrueForge's 0.183.0 line adds tenant-wide web search control and a reworked session-sharing model (per-user grants plus a tenant grant replacing the old shared-boolean semantics), alongside per-turn MCP tool injection, frontend Sentry tracing, and several sandbox and UI refinements.

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

01 Session sharing reworked: per-user grants and tenant grant

90

BREAKING

Sessions can now be shared with specific users when the `session_sharing.users` capability is reported, with invite search, grants, and revoke in the session access UI; tenant-wide access still goes through the session `shared` flag on `session.shared`, but tenant grants are now omitted from the session access collaborator list. The deprecated `PATCH /sessions/{id} { shared }` body field now grants or revokes a `tenant` grant instead of its previous shared-boolean behaviour, giving all tenant users read access by default — callers relying on the old sharing semantics must migrate to the `tenant` grant.

Grant all users in the tenant read access to a session so the whole team can view an ongoing conversation.

```
$ curl -X PATCH https://<host>/api/sessions/<session_id> \
  -H 'Content-Type: application/json' \
  -d '{"shared": true}'
```

— Names capability, fields, and endpoint with migration note and example

`@truefoundry/trueforge-ui@0.183.0` `@truefoundry/trueforge@0.183.0` `@truefoundry/trueforge-core@0.183.0`

02 Tenant-wide web search control via settings API NEW

85

Admins can view and toggle web search tenant-wide using `GET` and `PATCH` `/api/v1/settings/configuration`, gated to admins and backed by a new `configuration` table; session and agent capability writes respect the switch so web search cannot be enabled while it is turned off tenant-wide.

Read the current tenant-wide web search setting as an admin before deciding whether to enable it for all sessions.

```
$ curl -X GET https://<host>/api/v1/settings/configuration \
  -H 'Authorization: Bearer <admin-token>'
```

Inspect the current tenant-wide configuration, including web search status, before a deployment change.

```
$ curl -X GET https://<host>/api/v1/settings/configuration \
  -H 'Authorization: Bearer <admin-token>'
```

Read the current tenant-wide configuration, including web search status, as an admin.

```
$ curl -X GET https://<host>/api/v1/settings/configuration \
  -H 'Authorization: Bearer <admin-token>'
```

— Named endpoints, table, and gating mechanism with a runnable example

`@truefoundry/trueforge-ui@0.183.0` `@truefoundry/trueforge-assistant-ui-run…`

---

03 **Per-turn MCP tool injection via x-tfg-tools header** NEW

80

Pass `x-tfg-tools` as an array of objects (each with `inputSchema` and optional `outputSchema`) in a TrueFoundry-mode turn header to inject MCP tool definitions for a single turn without registering them globally.

Pass a custom MCP tool definition on a single turn without registering it globally.

```
$ curl -X POST https://<host>/api/v1/sessions/<session-id>/turns
\
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -H 'x-tfg-tools: [{"name": "lookup", "inputSchema": {"type":
"object", "properties": {"query": {"type": "string"}}},
"outputSchema": {"type": "string"}}]' \
  -d '{"message": "Look up the latest CVEs for OpenSSL"}'
```

Inject custom MCP tools for a single turn without reconfiguring the server — useful when a client wants to expose ephemeral or request-scoped tools.

```
$ curl -X POST https://<host>/api/turns \
  -H 'Content-Type: application/json' \
  -H 'x-tfg-tools: [{"name":"fetch_url","inputSchema":
{"type":"object","properties":{"url":
{"type":"string"}},"required":["url"]},"outputSchema":
{"type":"object"}}]' \
  -d '{"session_id": "<session_id>", "message": "Fetch
https://example.com/data.json and summarise it"}'
```

— Named header and schema with runnable curl examples `@truefoundry/trueforge@0.183.0` `@truefoundry/trueforge-core@0.183.0`

04 **Frontend Sentry error reporting and SSE tracing** NEW

65

Enable Sentry error reporting for the frontend by setting the `VITE_SENTRY_DSN` build argument; session-create and turn-stream API calls are then traced, recording SSE first-chunk latency, stall count, and stream errors as span attributes.

— Named build arg and traced metrics but no usage example `@truefoundry/trueforge@0.183.0`

THINNER COVERAGE BELOW

05 **422 responses for unowned catalog settings routes** BREAKING

50

Catalog settings routes that a deployment does not own now return `422` naming the route, where they previously returned `403`; callers that branched on `403` to detect unowned routes must be updated.

— Clear before/after status code but no concrete route path given

`@truefoundry/trueforge-ui@0.183.0` `@truefoundry/trueforge@0.183.0` `@truefoundry/trueforge-sdk@0.183.0` `@truefoundry/trueforge-assistant-ui-run…`

06 **(improved) Sandbox build and instruction improvements**

IMPROVED

50

Sandbox environment builds now fail fast on Daytona `4xx` responses, such as exceeding provider resource caps, with default resource caps lowered to 4 CPU / 8 GiB RAM / 10 GiB disk; sandbox instructions now also direct the model to save fetched remote content to a file before grepping or filtering it, rather than processing it directly.

— Concrete new default limits but no actionable command Brief behavioural note with no named surface

`@truefoundry/trueforge@0.183.0` `@truefoundry/trueforge-core@0.183.0`

07 **Settings capability reports per-section detail** IMPROVED

40

The `settings` capability now reports one entry per settings section, each with its own `enabled` and `manage_providers` fields.

— Names capability fields but no usage path shown `@truefoundry/trueforge-ui@0.183.0`

---

08 **Read-only environment drawer for non-manage users** NEW

35

Users without manage access can now open an environment in a read-only drawer to view its configuration.

— Describes the UI path but nothing deeper `@truefoundry/trueforge-ui@0.183.0`

---

09 **Configure model prompt when catalog is empty** IMPROVED

35

When the model catalog is empty, the UI now shows a Configure model button that opens Settings, replacing the previous Select model dropdown with a '?' monogram.

— Clear before/after UI change, no deeper mechanism `@truefoundry/trueforge-ui@0.183.0`

---

10 **Warning when sandbox disabled in agent runtime settings** NEW

35

Agent runtime settings now warn when the sandbox is turned off, noting that TrueForge will not work properly without it.

— Names the setting location but little mechanism `@truefoundry/trueforge-ui@0.183.0`

---

11 **Back control added after Try in header** NEW

20

A back control now appears in the header after Try, letting users return to the agents page.

— Minimal UI tweak with no further detail `@truefoundry/trueforge-ui@0.183.0`

└──▷ BREAKING ON UPGRADE

! Catalog settings routes that the deployment does not own now answer 422 (naming the route) where they previously answered 403.

! Catalog settings routes that the deployment does not own now answer 422 (naming the route) instead of 403.

! The deprecated `PATCH /sessions/{id} { shared }` body field now grants or revokes the `tenant` grant rather than its previous behaviour — callers relying on the old sharing model must migrate to the `tenant` grant.

! Catalog settings routes that the deployment does not own now return 422 instead of 403; callers that branch on 403 for unowned routes must be updated.

! Session grants now give read access by default; the semantics of `PATCH /sessions/{id} { shared }` have changed — it now grants or revokes a `tenant` grant rather than the previous `shared` boolean behaviour, which is deprecated.

! Catalog settings routes that the deployment does not own now return 422 (naming the route) instead of 403 — callers that check for 403 to detect unowned routes must be updated.

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

kordoc

Feedback ⓘ v4.24.42 NOTES ↗ RANK

모두 파싱해버리겠다 — HWP·HWPX·PDF·Office 문서를 Markdown으로. 양식 자동 채우기와 신구 대조를 갖춘 CLI·MCP 서버 | Convert Korean documents (HWP, HWPX, PDF, Office) to Markdown — CLI and MCP server with form filling and diff

> kordoc's OCR pipeline gained a way to pin recognition to the default printed-text model and skip handwriting downloads, plus automatic fallback to the handwriting model when confidence is low.

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

01 Force printed-only OCR recognition model NEW

83

Setting `ocrRec: "printed"` in config, `--ocr-rec printed` on the CLI, or `ocr_rec` in MCP pins OCR to the default recognition model only, skipping handwriting model downloads entirely — useful in CI or air-gapped environments processing only printed documents.

Prevent the ~15 MB handwriting model from being downloaded or used in a CI or air-gapped environment where only printed documents are processed.

```
$ npx kordoc 스캔본.pdf --ocr --ocr-rec printed -o 스캔본.md
```

— Names exact flag, config key, MCP param, and runnable CLI example. v4.24.42

02 Automatic handwriting model fallback by confidence IMPROVED

60

kordoc's default OCR path now automatically switches to the handwriting recognition model on pages where confidence is below .85 and switching would raise the score by more than .10, without requiring any configuration change.

— Explains exact thresholds but no config surface to act on. v4.24.42

WAS THIS USEFUL? 👍 👎

tabularis

Feedback ⓘ v0.31.0 NOTES ↗ RANK

Open-source desktop SQL workspace with 3 built-in database drivers and 24 shipped driver plugins, including SQL Server, DuckDB, ClickHouse and Redis. Built-in MCP server for Claude, Cursor and Devin, SQL notebooks and visual EXPLAIN.

> Tabularis v0.31.0 adds connection import from HeidiSQL, MySQL Workbench and Navicat, a transposed result-grid layout, in-grid text search, and a cluster of new SQL export and copy options across saved queries, query history, and the grid context menu.

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

01 Connection import from HeidiSQL, Workbench, Navicat NEW

75

Import existing connections from HeidiSQL, MySQL Workbench, and Navicat `.ncx` export files via the new connection importer, avoiding re-entering credentials.

Bring existing connections from Navicat into tabularis without re-entering credentials.

> 📍 In the Connections panel, choose Import connections, select Navicat, and pick your `.ncx` export file.

— Names three source tools and file format, with UI steps v0.31.0

THINNER COVERAGE BELOW

02 Transposed result grid layout NEW

55

The result grid now supports a transposed layout that flips rows and columns, intended for inspecting wide-schema tables.

Inspect a wide table with many columns by flipping rows and columns in a transposed view.

> 📍 In the result grid, right-click the grid header area and select Transposed layout to toggle the row/column orientation.

— UI action given, no deeper mechanism described v0.31.0

03 Context-menu copy and export formats NEW

53

Copy rows as SQL UPDATE statements directly from the grid context menu, and the Export Selected context menu gains Markdown and SQL INSERT output formats.

— Names context menu and formats, no exact steps v0.31.0

04 **In-grid text search in result data** NEW 45

A new in-grid text search lets you find and highlight text across already-loaded result rows without writing a new query.

Locate a specific value in a large result set without writing a new query.

> 📍 In the data grid, open the in-grid text search (find text in loaded results) and type the value to highlight matching cells.

— Clear UI path but no technical depth v0.31.0

---

05 **SQL export for query history and saved queries** NEW 45

Export filtered query history as SQL, and export all saved queries or an entire folder of saved queries to a single `.sql` file.

— Two export surfaces named, no exact UI steps v0.31.0

---

06 **Pin tables and views in sidebar** NEW 34

Pin tables and views to the top of the sidebar explorer list for faster navigation.

— Named feature, screenshot but no steps v0.31.0

---

07 **Index and foreign key details in sidebar** NEW 34

The sidebar panel now shows index and foreign key details for tables.

— Thin addition, no navigation detail given v0.31.0

---

08 **Hide and show grid columns** NEW 28

Hide and show individual columns in the current result grid without changing the underlying query.

— Thin one-line description, no navigation given v0.31.0

└──▷ ALSO FROM THESE RELEASES



WAS THIS USEFUL? 👍 👎

api-mcp-server

Feedback ⓘ v3.27.0 NOTES ↗ RANK

MCP server that exposes Hostinger's platform API for hosting deployment, CDN control, domain management, and WordPress operations.

> api-mcp-server expands its GPU catalog with four new query operations covering pool groups, instance types, OS templates, and full pool listings.

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

01 **GPU catalog query operations** NEW 67

Four new operations let callers query GPU catalog data: `gpu_catalog_pool-groups` for available GPU pool groups, `gpu_catalog_pool-instance-types` for available GPU pool instance types, `gpu_catalog_pool-os-templates` for available GPU pool OS templates, and `gpu_catalog_pool` for the full GPU pool list.

— Names all four operations but no usage detail beyond that. v3.27.0

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ AI / LLM Gateways

Mozilla AI Otari

Feedback ⓘ v0.37.0 NOTES ↗ RANK

Open-source, OpenAI-compatible LLM gateway you run yourself. One endpoint for 40+ providers, with virtual keys, budgets, and usage tracking.

> Across v0.36.0 and v0.37.0, Otari adds a direct code-execution API endpoint with tool-call reporting, per-workspace web search provider key management, and individual pricing for search and fetch tools, while also improving the model catalog and automatic routing view and removing the learned kNN and noop routers.

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

01 Code execution endpoint and tool-call reporting NEW

90

Call `POST /api/v1/code/execute` to run code directly without routing a request through a model. Chat responses now also report code execution and MCP calls via the `otari_tool_calls` field in the response.

Run a code snippet directly against the sandbox without involving a model — useful for CI pipelines that need isolated execution.

```
$ curl -X POST https://<your-otari-host>/api/v1/code/execute \
  -H 'Authorization: Bearer <api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"language": "python", "code": "print(1 + 1)"}'
```

Inspect tool calls (code execution and MCP calls) that occurred during a chat turn by reading the `otari_tool_calls` field in the response.

```
$ curl -X POST https://<your-otari-host>/v1/chat/completions \
  -H 'Authorization: Bearer <api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "auto", "messages": [{"role": "user", "content":
"What is 2+2? Use code."}]}' \
  | jq '.otari_tool_calls'
```

— Exact endpoint, field name, and runnable curl examples given. v0.36.0

02 Web search provider key management per workspace NEW

75

Organizations can add, edit, and assign web search provider keys per workspace through `Tools > Web search` in the dashboard, and workspaces can pin which provider runs web search and fetch, overriding organization-level defaults; workspaces fall back to the deployment's tool when the organization has no default configured.

Set up an organization's web search keys and assign them to a workspace so that workspace uses a specific search provider.

> 📍 In the dashboard, go to Organization › Web search providers, add a provider key, then go to the target workspace, open Tools › Web search, and select the key.

— Exact UI path and walkthrough example given, full behavior described. v0.36.0

---

03 **Individual pricing for search and fetch tools** BREAKING

75

Search and fetch tools now each carry their own price instead of a shared one, so the dashboard shows per-tool costs and pricing routes accept every price unit; any configuration relying on shared or inherited pricing will produce different costs after upgrading. Stored search and fetch tools with no price assigned now fall back to the active Otari platform price instead of being unpriced, and pricing rates are cached per process for 30 seconds to reduce load on the pricing backend.

— Describes mechanism and before/after but no exact config keys shown. v0.37.0

---

04 **Model catalog v2 with kind-aware health checks** IMPROVED

65

The catalog now fetches the `models.dev` v2 catalog and falls back to v1 automatically, and catalog models now carry a `kind` field (chat, embedding, etc.), with health checks consulting a model's kind before probing it as chat.

— Names catalog source and field but gives no reader action. v0.36.0

THINNER COVERAGE BELOW

05 **Learned kNN and noop routers removed** BREAKING

55

The learned kNN and noop routers have been removed from the routing layer; any setup that referenced either router will break on upgrade.

— Clear breaking change but no migration path given. v0.37.0

06 **Operator email and domain blocking for sign-in** NEW

50

Operators can now block specific email addresses and domains from signing in via the admin panel.

— Names admin panel location but no exact field or flag. v0.36.0

07 **Automatic fallback routing view improvements** IMPROVED

45

The Automatic fallback routing view now shows catalog order with prices and which configuration named each provider, alongside improvements to automatic model selection and sticky sessions in routing.

— UI change described but no exact navigation or config names. v0.36.0

08 **Serply adapter for any-search provider** NEW

45

A new `Serply` adapter has been added to the `any-search` provider, giving operators another web search integration alongside existing providers.

— Names the adapter and provider but no usage shown. v0.37.0

09 **User-Agent header on search and fetch providers** NEW

45

Every `any-search` and `any-fetch` provider now sends a `User-Agent` header that names the library in outbound requests.

— Names the header but gives no further mechanism. v0.37.0

10 **Hugging Face keys billed to organization budget** NEW

40

Hugging Face provider keys can now bill against an organization's budget.

— Single-line addition with no further mechanism given. v0.36.0

11 **Budget name and limits in end-user responses** IMPROVED

40

End-user budget responses now return each budget's name and limits.

— Minimal description, no endpoint or example provided. v0.36.0

└──▷ **BREAKING ON UPGRADE**

! Search and fetch tools are now priced by their own individual price only — any configuration that relied on a shared or inherited price will produce different costs after upgrading.

! The learned kNN and noop routers have been removed from the routing layer — any setup that referenced either router will break on upgrade.

WAS THIS USEFUL? 👍 👎

BerriAI LiteLLM

Feedback ⓘ v1.106.7 NOTES ↗ RANK

LiteLLM is an AI gateway that calls over 100 model APIs in OpenAI format with cost tracking, guardrails, load balancing, and logging.

> LiteLLM's latest releases add video generation routing, Bing Search grounding and Vertex AI Interactions API support, a Gemini Family auto-router preset, and router/auth improvements, while removing the bundled Lens dashboard and deleting the prompt_token_calculator utility.

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

01 **Video generation via custom provider map** NEW 65

Serves `custom_provider_map` `BaseVideoConfig` providers on the `/v1/videos` endpoint, enabling custom video provider routing through the proxy.

— Names endpoint and config class but no usage example v1.106.7

02 **Azure AI Foundry auth and JWT route wildcards** IMPROVED 60

Improves auth by supporting Entra ID and OAuth on every Azure AI Foundry route, and supporting wildcard prefixes in JWT `team_allowed_routes`.

— Names auth methods and config field, no example shown v1.108.0

THINNER COVERAGE BELOW

03 **Complexity router reasoning-effort controls** IMPROVED 55

Improves the router by adding per-group supported reasoning efforts with a maximum level cap, and bounds the classifier context block in the complexity router to control token usage.

— Explains mechanism but no config key or command v1.108.0

04 **Bing Search grounding and Vertex AI Interactions API** NEW 50

Adds `bing_grounding` as a named search provider for Grounding with Bing Search, and adds native Vertex AI Interactions API support.

— Names provider and API but no mechanism detail v1.108.0

05 **Gemini Family auto-router preset in UI** NEW 45

Adds a Gemini Family auto-router preset in the UI for one-click Gemini model routing.

— UI location given but no further mechanism v1.108.0

06 **prompt_token_calculator utility deleted** BREAKING 45

The `prompt_token_calculator` utility function has been deleted; any code that imported or called it directly will break.

— Names function but gives no migration path v1.108.0

07 **Error-code drilldown on UI caching page** NEW 40

Adds error-code drilldown for failed requests on the UI caching page.

— Clear UI location, thin on mechanism v1.108.0

08 **Together AI serverless models added to cost map** NEW 35

Adds missing Together AI serverless models to the cost map.

— Names provider but no model list or numbers v1.108.0

09 **Lens dashboard removed from bundle** BREAKING 25

The Lens dashboard is no longer bundled or built starting in the 1.106.x release line.

— States removal only, no migration guidance v1.106.7

└──▷ **BREAKING ON UPGRADE**

! The Lens dashboard is no longer bundled or built in the 1.106.x release line.

! The `prompt_token_calculator` utility function has been deleted; any code that imported or called it directly will break.

WAS THIS USEFUL?  

litellm-rs

Feedback ⓘ v0.10.0 NOTES ↗ RANK

litellm-rs is a Rust port of LiteLLM, providing a unified API gateway that routes requests across many LLM providers with load balancing and fallbacks.

> litellm-rs v0.10.0 adds cost tracking to usage data, with OpenRouter now settling charges from provider-reported cost, though the schema change breaks existing struct literals.

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

01 Cost field in Usage struct with OpenRouter settlement

83

BREAKING

Adds a public `cost: Option<f64>` field to `litellm_rs::core::types::responses::Usage`, populated from upstream provider responses. OpenRouter chat completions now settle costs from OpenRouter's reported `usage.cost`, with the request ledger recording `charge_basis = "provider_reported"`. Code that builds `Usage { .. }` with a struct literal must now add `cost: None`, or use `Usage::new` or ..Default::default(), since the new field has no default in struct-literal position.

— Names exact field, type, provider, and migration fix needed. v0.10.0

└──▷ **BREAKING ON UPGRADE**

! Code that builds `Usage { .. }` with a struct literal must now add `cost: None`, or use `Usage::new` or ..Default::default(), because the new `cost: Option<f64>` field has no default in struct-literal position.

WAS THIS USEFUL? 👍 👎

SMG Project Shepherd Model Gateway Feedback ⓘ

v1.14.0 NOTES ↗

Shepherd Model Gateway is an engine-agnostic LLM gateway in Rust with full OpenAI & Anthropic API compatibility, gRPC pipeline, and KV cache routing.

SMG adds a separate admin control plane, secrets-off-CLI credential handling, Symphony metrics, ZMQ timeout flag, DP-aware routing, and per-role readiness counts.

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/smg-project/smg:1.14.0
```

pipx

```
$ pipx install smg==1.14.0
# already installed? upgrade to this version:
$ pipx install --force smg==1.14.0
```

Cargo

```
$ cargo install smg --version 1.14.0
# already installed? upgrade to this version:
$ cargo install smg --version 1.14.0 --force
```

git

```
$ git clone --branch v1.14.0 https://github.com/smg-project/smg.git
# already have the repo? check out this version:
$ git checkout v1.14.0
```

published under the project's GitHub account · named in the project's README · checked 4 Nov

Wrong command?

Serve a separate admin, worker-registry, and RL control plane with `--admin-port`; bound the data listener with `--max-connections` and snapshot in-flight requests with `GET /inflight`.

Suppress credential exposure by passing the API key via `SMG_API_KEY`, per-tenant keys via `SMG_TENANT_API_KEYS`, database URLs via `SMG_POSTGRES_DB_URL` and

`SMG_REDIS_URL`, or file paths via `--api-key-file` and `*_FILE` environment variables.

Monitor Symphony output health with new metrics `smg_symphony_outputs_total`, `smg_symphony_malformed_total{reason}` (28 reasons), and `smg_symphony_refused_total`.

Expose a `/metrics` endpoint on every engine wire with `--metrics-port`, reporting finish-reason breakdowns, refusals, errors, in-flight counts, token counts, TTFT, inter-token, and end-to-end histograms.

Track request latency with two new log-spaced Prometheus histograms: `smg_router_time_to_first_token_seconds` and `smg_router_request_time_per_output_token_seconds`.

+9 more

NEW DETECTION Symphony parser tables for new model families

NEW SUPPORTED PLATFORM TensorRT-LLM Rust gRPC servicer support

UPDATED SUPPORTED PLATFORM SGLang Rust servicer gains disaggregation and multimodal support

└─▷ **BREAKING ON UPGRADE**

- ! Symphony is now the default parser; deployments relying on the previous default parsers must pass `--parser legacy` or send `X-SMG-Parser: legacy` to restore the old behavior.
- ! `--kv-index`, `--kv-indexer-ttl-secs`, and `--kv-indexer-max-entries` are removed; a config that still sets any of them will not start.
- ! Per-policy `load_check_interval_secs` is refused; replace it with `load_monitor_interval_secs`.
- ! The gateway's API key is no longer forwarded upstream; workers and external providers that relied on it need their own `api_key` at registration.
- ! Engine pods take longer to become Ready because servicers now warm up before reporting SERVING; raise startup probes and rollout deadlines, and note that the Helm chart's `router.progressDeadlineSeconds` is now 1920 s.
- ! `tool_choice: 'none'` now renders tools in the prompt (matching engine server behavior); prompt token counts and prefix-cache keys for affected requests change on upgrade.
- ! Helm: the API key, Postgres and Redis URLs, and Oracle credentials must now come from Secrets (`auth.existingSecret`, `history.<store>.existingSecret`); worker images default to `ghcr.io/smg-project/smg`; discovery providers must be configured under `router.discovery`.
- ! Python bindings: Router(...) and RouterArgs(...) now require every parameter after (worker_urls, policy, host, port) and (worker_urls, host, port) respectively to be

passed as keyword-only arguments.

- ! Six metric families are removed and the error counter gains a mandatory `reason` label; dashboards and alerts targeting the removed families or the unlabelled counter will break.
- ! Upgrade order is required: roll gateways first (SMG 1.14.0), then engine pods with `smg-grpc-servicer` 0.15.0.

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

FlashInfer

Feedback ⓘ v0.7.3 NOTES ↗ RANK

FlashInfer provides GPU kernels for language-model inference, including attention, sampling, and quantized operations.

> FlashInfer v0.7.3 unifies MLA backend execution with autotuning, expands the unified MoE API with cuDNN grouped-GEMM backends and a wide set of new MoE kernels (Rubin, TRTLLM-GEN, DeepSeek-V4-Flash, SM120 MegaMoE), and adds MiniMax-H3 MLP blocks plus several SM120 kernels and Cake-backend attention paths for RTX 50-series GPUs.

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

01 **New MoE kernels for Rubin, TRTLLM-GEN and DeepSeek-V4-Flash**

NEW 75

Adds Rubin (SM107) locality-domain execution enabling expert-parallel workloads on Rubin-class GPUs, TRTLLM-GEN FP8 routed MoE on SM107 (Rubin) GPUs, TRTLLM Gen SwiGLU StepFun support in the MoE kernel suite, a fused tanh-clamped squared ReLU in CUTLASS MoE epilogues, clamped DeepSeek-V4-Flash routed NVFP4 decode for SM100 GPUs, and an SM120 W4A8 CuTeDSL MegaMoE backend for RTX 50-series and DGX Spark GPUs.

— Names six specific MoE kernel surfaces and target GPUs, no code example v0.7.3

02 **MiniMax-H3 full MLP block operators** NEW 75

Adds MiniMax-H3 full MLP block operators: a BF16, MXFP8, and NVFP4 full MLP block for SM100 and SM103, and an SM120 fused quantized full MLP (FC1 + SwiGLU + FC2 + gated residual) for RTX 5090 and RTX PRO 6000.

Run MiniMax-H3 varlen attention with NVFP4 quantization on packed sequences — useful for multi-segment prefill on Blackwell hardware.

```python
from flashinfer.prefill import minimax_h3_varlen_nvfp4_attention
import torch

cu_seqlens = torch.tensor([0, 133, 300, 900], dtype=torch.int32,
device='cuda')
T, heads = 900, 7
q = torch.randn((T, heads, 128), dtype=torch.bfloat16,
device='cuda')
k, v = torch.randn_like(q), torch.randn_like(q)

# pv_mode='fp8' or 'fp4'
out = minimax_h3_varlen_nvfp4_attention(q, k, v, cu_seqlens,
pv_mode='fp8')
```

— Names precisions, SMs and GPUs plus a related runnable example v0.7.3

03

Cake backend kernels for MLA K/V, NVFP4 attention and Kimi-K3 routing

NEW

65

Generates Cake kernels for the fused MLA K/V pack and FP8 cast via `cake_concat_mla_kv_quant_fp8`, integrates Cake-backend NVFP4 attention reachable through prepare_nvfp4_attention(..., backend='cake'), and adds `cake_kimi_k3_attn_res` serving-shape routing covering sequence lengths M 1 to 16384 plus a snapshot-write variant for SM100 and SM103.

Use NVFP4 attention on a Blackwell GPU with the Cake backend via `prepare_nvfp4_attention`.

```python
from flashinfer.prefill import prepare_nvfp4_attention
import torch

q = torch.randn((4, 8, 4096, 128), dtype=torch.bfloat16,
device='cuda')
k, v = torch.randn_like(q), torch.randn_like(q)
output = torch.empty_like(q)
attention = prepare_nvfp4_attention(q, k, v, output,
backend='cake')
attention()
```

— Names function and has a runnable example Names function and range but no usage example

v0.7.3

04 **Auto backend selection for single-token d256 paged decode**

IMPROVED 60

`backend='auto'` now selects cuDNN for single-token d256 paged decode, with CUDA-graph wrappers declaring replay compatibility to cudnn-frontend 1.31+.

— Names flag value, dimension and version requirement v0.7.3

THINNER COVERAGE BELOW

05 **Minimal mode for architecture-targeted installs** IMPROVED

55

`install-jit-cache-wheel --sm` now implies minimal mode so architecture-targeted installs skip unneeded kernel variants.

— Exact flag named, directly runnable v0.7.3

06 **Unified MLA backend with wrapper autotuning** IMPROVED

50

FlashInfer unifies MLA backend execution and adds wrapper autotuning via the updated MLA API, letting the library automatically pick the best backend per hardware.

— Names updated MLA API but no flag or runnable example v0.7.3

07 **Unified MoE API gains cuDNN grouped-GEMM backends** IMPROVED

50

The unified MoE API is expanded with cuDNN grouped-GEMM backends and the remaining Prims-TS quantization pairs.

— Names backends and quantization pairs, no usage shown v0.7.3

08 **SM120 paged FP8 MQA lightning-indexer logits** NEW 40

Adds SM120 paged FP8 MQA lightning-indexer logits for RTX PRO 6000 and RTX 5090, described as an experimental generated-program export.

— Names hardware and kernel type but no usage path v0.7.3

09 **cuDNN linear attention pools and bounded auto dispatch** NEW

25

Integrates cuDNN linear attention pools and bounded auto dispatch.

— Bare description with no mechanism or usage v0.7.3

WAS THIS USEFUL? 👍 👎

◆ Agentic Infrastructure

ZenML

Feedback ⓘ 0.98.1 NOTES ↗ RANK

ZenML is an MLOps and LLMOps framework for building portable pipelines that run the same code across local, cloud, and orchestrator backends.

> ZenML 0.98.1 adds a CLI command for renaming projects independently of their technical identifiers, alongside hardened default Helm chart pod security settings.

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

01 **Project renaming via CLI** NEW 88

Projects can now be renamed from the CLI using `zenml project update <name-or-uuid>` with the `--name`, `--display-name`, and `--description` flags, allowing the technical name and display name to be changed independently while preserving the project UUID and all associated resources.

Rename a project's display name without touching its technical name, keeping all associated pipeline runs and resources intact.

```
$ zenml project update my-old-project --display-name 'Fraud Detection v2' --description 'Renamed for Q3 reorg'
```

— Exact command and flags with behavior and preserved state 0.98.1

THINNER COVERAGE BELOW

02 **Restricted Pod Security Standard compliance in Helm chart**

IMPROVED 50

ZenML's default Helm chart pod security settings now satisfy Kubernetes Pod Security `restricted` enforcement, covering CA certificate init containers and Helm test pods.

— Names the standard and affected components but no config steps 0.98.1

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.50.0 NOTES ↗

RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> The OpenAI TypeScript SDK's biggest update is keyless workload identity authentication for Kubernetes, Azure, GCP and X.509/mTLS clients, alongside new Responses API caching and WebSocket output retention improvements.

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

01 Workload identity authentication for keyless auth NEW

95

Clients can now authenticate without long-lived API keys by passing a `workloadIdentity` object with `identityProviderId`, `serviceAccountId`, and `provider` to the `OpenAI` constructor. Built-in providers `k8sServiceAccountTokenProvider`, `azureManagedIdentityTokenProvider`, and `gcpIDTokenProvider` from `openai/auth` cover Kubernetes, Azure, and GCP; `refreshBufferSeconds` on `workloadIdentity` (default 1200 seconds, capped at half the token lifetime) controls proactive token refresh timing. X.509 client certificate authentication is supported via `workloadIdentity.fromX509` from `openai/auth/x509-transport`, taking `certificateChain`, `privateKey`, `identityProviderId`, and `serviceAccountId` and targeting `https://mtls.api.openai.com/v1` (Node.js only, requires the `undici` peer dependency). Custom subject-token providers can also be supplied as an object with `tokenType` and `getToken` passed as the `provider` field.

Authenticate from a Kubernetes pod using a service account token instead of an API key.

typescript

```typescript
import OpenAI from 'openai';
import { k8sServiceAccountTokenProvider } from 'openai/auth';

const client = new OpenAI({
  workloadIdentity: {
    identityProviderId: 'idp-123',
    serviceAccountId: 'sa-456',
    provider:
k8sServiceAccountTokenProvider('/var/run/secrets/kubernetes.io/se
rviceaccount/token'),
    refreshBufferSeconds: 120,
  },
});

const response = await client.chat.completions.create({
  model: 'gpt-5.5',
  messages: [{ role: 'user', content: 'Hello!' }],
});
```

Authenticate from an Azure workload using managed identity, with no API key required.

typescript

```typescript
import OpenAI from 'openai';
import { azureManagedIdentityTokenProvider } from 'openai/auth';

const client = new OpenAI({
  workloadIdentity: {
    identityProviderId: 'idp-123',
    serviceAccountId: 'sa-456',
    provider: azureManagedIdentityTokenProvider(),
  },
});

const response = await client.chat.completions.create({
  model: 'gpt-5.5',
  messages: [{ role: 'user', content: 'Hello!' }],
});
```

Authenticate using an X.509 client certificate over mTLS instead of an API key (Node.js only).

```typescript
import OpenAI from 'openai';
import { workloadIdentity } from 'openai/auth/x509-transport';

const credential = workloadIdentity.fromX509({
  certificateChain:
process.env['OPENAI_X509_CLIENT_CERTIFICATE_CHAIN_PEM']!,
  privateKey: process.env['OPENAI_X509_CLIENT_PRIVATE_KEY_PEM']!,
  identityProviderId:
process.env['OPENAI_X509_IDENTITY_PROVIDER_ID']!,
  serviceAccountId:
process.env['OPENAI_X509_SERVICE_ACCOUNT_ID']!,
});

try {
  const client = new OpenAI({
    credential,
    project: process.env['OPENAI_X509_PROJECT_ID'] ?? null,
  });
  console.log((await client.models.list()).data.length);
} finally {
  await credential.close();
}
```

— Names every config key and provider, with three runnable code examples v7.49.0

THINNER COVERAGE BELOW

02 **Expanded Live Responses delegation tools and choices** BREAKING

45

Live Responses delegation tools and choices have been expanded with new variants. This is a breaking change for TypeScript consumers using exhaustive `switch` statements over delegation tools or choices, who may need to add new variant cases to remain type-complete.

— Breaking-change impact named but specific new variants unnamed v7.48.0

03 **12-hour caching for the Responses API** NEW

40

The Responses API now supports 12-hour caching, and callers can read cache hits and cache-write usage data from Responses API calls. The release also mentions project-level service tier settings alongside this caching work.

— No code example, flag, or config key given for either change v7.50.0

04 **Sparse WebSocket output retention in Responses API** IMPROVED

The Responses API now retains sparse WebSocket output facts, so callers receive complete output items even when only partial data was emitted over the connection.

— Mechanism described in prose only, no API or field named `v7.48.0`

└──▷ BREAKING ON UPGRADE

! Exhaustive TypeScript `switch` statements over Live Responses delegation tools or choices may need new variant cases to remain type-complete after this expansion.

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.45.0 NOTES ↗ RANK

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

> OpenAI Python SDK's biggest addition this window is workload identity authentication for Kubernetes, Azure, GCP and X.509 mTLS environments, alongside new Responses caching with usage tracking and incremental improvements to Live Responses streaming.

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

01 Workload identity authentication for cloud environments NEW

95

Authenticate without long-lived API keys by passing a `workload_identity` dict to OpenAI(), with `identity_provider_id`, `service_account_id`, and a `provider` drawn from `openai.auth` helpers `k8s_service_account_token_provider`, `azure_managed_identity_token_provider`, or `gcp_id_token_provider`. Token refresh timing is controlled via `refresh_buffer_seconds` inside the `workload_identity` dict (default 1200 seconds before expiry). X.509 mutual TLS workload identity is supported via openai.auth.x509_workload_identity() combined with a configured `DefaultHttpx2Client` and TLS context, defaulting to `https://mtls.api.openai.com/v1` and handling token exchange, caching, and refresh automatically; a custom short-lived token source can be supplied by setting `provider` to a dict with `token_type` and a `get_token` callable. The same federation flows work asynchronously via `AsyncOpenAI` and `DefaultAsyncHttpx2Client`.

Authenticate a Kubernetes pod to OpenAI using its projected service account token instead of a static API key.

```python
from openai import OpenAI
from openai.auth import k8s_service_account_token_provider

client = OpenAI(
    workload_identity={
        "identity_provider_id": "idp-123",
        "service_account_id": "sa-456",
        "provider": k8s_service_account_token_provider(
            "/var/run/secrets/kubernetes.io/serviceaccount/token"
        ),
        "refresh_buffer_seconds": 120.0,
    },
)

response = client.chat.completions.create(
    model="gpt-5.5",
    messages=[{"role": "user", "content": "Hello!"}],
)
print(response.choices[0].message.content)
```

Authenticate via X.509 mTLS in a zero-secret environment where the workload certificate is provisioned by a certificate authority.

```python
import os, ssl
from openai import OpenAI, DefaultHttpx2Client
from openai.auth import x509_workload_identity

tls_context =
ssl.create_default_context(cafile=os.getenv("OPENAI_MTLS_CA_BUNDL
E"))
tls_context.load_cert_chain(
    certfile=os.environ["OPENAI_MTLS_CERTIFICATE_CHAIN"],
    keyfile=os.environ["OPENAI_MTLS_PRIVATE_KEY"],
)

client = OpenAI(
    workload_identity=x509_workload_identity(

identity_provider_id=os.environ["OPENAI_IDENTITY_PROVIDER_ID"],

service_account_id=os.environ["OPENAI_SERVICE_ACCOUNT_ID"],
    ),
    http_client=DefaultHttpx2Client(verify=tls_context,
follow_redirects=False),
)

response = client.chat.completions.create(
    model="gpt-5.5",
    messages=[{"role": "user", "content": "Hello!"}],
)
print(response.choices[0].message.content)
```

Authenticate a GCP Compute Engine workload to OpenAI using a metadata-server-issued ID token.

```python
from openai import OpenAI
from openai.auth import gcp_id_token_provider

client = OpenAI(
    workload_identity={
        "identity_provider_id": "idp-123",
        "service_account_id": "sa-456",
        "provider":
gcp_id_token_provider(audience="https://api.openai.com/v1"),
    },
)

response = client.chat.completions.create(
    model="gpt-5.5",
    messages=[{"role": "user", "content": "Hello!"}],
)
print(response.choices[0].message.content)
```

— Names exact dict keys, helper functions, defaults, and runnable code v3.44.0

THINNER COVERAGE BELOW

02 **12-hour Responses caching with cache-write usage tracking**

NEW

30

The SDK now supports 12-hour Responses caching and exposes cache-write usage data in the API.

— States the capability but no config, endpoint, or usage detail v3.45.0

03 **Live Responses streaming and delegation improvements** IMPROVED

30

The responses API now accumulates richer WebSocket progress events for real-time streaming state, adds WebSocket lane overflow isolation to keep concurrent streams from interfering with each other, and expands the delegation tools available to Live Responses.

— Describes three thin improvements with no named flags or endpoints v3.43.0

WAS THIS USEFUL? 👍 👎

Anyscale Ray

Feedback ⓘ ray-2.60.0 NOTES ↗ RANK

Ray is a distributed compute engine for AI workloads, bundling Ray Train, Ray Serve, Ray Data, and Ray Tune on a common runtime.

> Ray 2.60 adds new Ray Data write formats and introspection tooling, surge-based rolling updates and gRPC reflection in Ray Serve, and a rate-based autoscaler for Serve metrics, alongside two breaking changes to how Serve handles autoscaling metrics and `root_path`.

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

01 **Surge rolling updates in Ray Serve** NEW 75

A new `max_surge_percent` deployment option keeps rolling updates at full capacity by starting replacement replicas before stopping old ones; it defaults to `0`, preserving the existing stop-then-start behavior.

Keep serving capacity stable during a rolling deployment update by starting new replicas before stopping old ones.

python

```python
from ray import serve
from ray.serve.config import DeploymentConfig

@serve.deployment(max_surge_percent=25)
class MyModel:
    def __call__(self, request):
        return 'ok'

serve.run(MyModel.bind())
```

— Named flag, default, and runnable code example ray-2.60.0

02 **Ray Data write formats, schema and introspection additions**

NEW 72

Ray Data gained Dataset.write_orc() to write ORC files, a `min_bytes_per_file` option in `write_parquet` to combine small output blocks toward a target file size, nested schema evolution in `write_delta` via the `schema_mode` parameter, the new ray.data.list_stats_summaries() developer API to query execution stats of completed datasets, and a new unnest() expression to expand struct-valued columns into one column per field.

Write a processed dataset to ORC format for downstream consumers that prefer columnar storage.

```python
import ray

ds = ray.data.read_parquet('s3://my-bucket/input/')
ds.write_orc('s3://my-bucket/output-orc/')
```

Inspect execution stats of recently completed datasets to diagnose pipeline bottlenecks.

```python
import ray.data

summaries = ray.data.list_stats_summaries()
for s in summaries:
    print(s)
```

— Five named APIs/params with examples for two `ray-2.60.0`

---

03 **gRPC server reflection in Ray Serve** NEW 70

Ray Serve can now enable gRPC server reflection so tools like `grpcurl`, `grpcui`, and Postman can discover services without local proto files; it can be disabled with gRPCOptions(enable_reflection=False) or the `--grpc-disable-reflection` flag.

— Names the tools and exact disable flags, no example `ray-2.60.0`

---

04 **Controller-side autoscaling metrics aggregation** BREAKING 65

Ray Serve now always aggregates autoscaling metrics at the controller, removing the `RAY_SERVE_AGGREGATE_METRICS_AT_CONTROLLER` flag and changing default behavior for anyone who had not explicitly enabled it; the related columnar autoscaling metrics format is now on by default, cutting controller loop cost 2.8x–13.5x at scale, and can be reverted with `RAY_SERVE_COLUMNAR_AUTOSCALING_METRICS=0`.

— Names both env vars and a measured perf gain `ray-2.60.0`

---

05 **ASGI-compliant root_path handling** BREAKING 60

`root_path` now follows the ASGI spec in every ingress mode; requests that still include the `root_path` prefix return 404, so operators must strip it at their ingress or unset `root_path`. The minimum Uvicorn version is now 0.26, with the bundled version upgraded to 0.53.

— Clear migration note and version numbers given `ray-2.60.0`

THINNER COVERAGE BELOW

06 **Node compaction for pack scheduling in Ray Serve** NEW

55

Node compaction for pack scheduling can be enabled with `RAY_SERVE_USE_PACK_SCHEDULING_STRATEGY=1`, with timing tunable via `RAY_SERVE_NODE_COMPACTION_DELAY_S`.

— Two env vars named, no deeper mechanism given `ray-2.60.0`

07 **Prometheus-based custom autoscaling policies** NEW 45

A new alpha `PrometheusQueryMixin` lets Ray Serve scale on Prometheus metrics for custom autoscaling policies, keeping PromQL query results fresh via a background thread.

— Alpha feature named but no usage example `ray-2.60.0`

└──▷ **BREAKING ON UPGRADE**

! Autoscaling metrics in Ray Serve are now always aggregated at the controller; the `RAY_SERVE_AGGREGATE_METRICS_AT_CONTROLLER` flag is removed, changing the default behavior for anyone who had not explicitly enabled it.

! `root_path` now follows the ASGI spec in every ingress mode; requests that still include the `root_path` prefix return 404 — strip it at your ingress or unset `root_path`. Uvicorn minimum version is now 0.26 (upgraded to 0.53).

WAS THIS USEFUL? 👍 👎

ArcBox

Feedback ⓘ v0.11.2 NOTES ↗ RANK

ArcBox runs containers, microVM agent sandboxes, and full Linux or macOS virtual machines on a Mac from one Rust daemon and CLI.

> ArcBox added a QEMU-based fallback backend for its System VM on Apple silicon, complete with software emulation, host network connectivity, and live machine control.

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

01 **QEMU backend for System VM** NEW 73

ArcBox can now boot the System VM on QEMU as a fallback on Apple silicon hosts where VZ or HV hardware virtualization is unavailable, using QEMU TCG (software emulation). It auto-discovers an installed QEMU build when no QEMU path is configured, opens a QMP (QEMU Machine Protocol) session for live machine control, and serves virtio-fs and vsock to the QEMU machine over vhost-user while attaching NIC1. The QEMU System VM also gained a vmnet bridge NIC, giving containers and VMs running under the QEMU backend direct host-to-VM network connectivity.

— Mechanism and components named in detail, but no user-facing flag or command. v0.11.1 v0.11.2

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

Comfy Org ComfyUI

Feedback ⓘ v0.44.0 NOTES ↗ RANK

ComfyUI is a node-graph interface, API, and backend for running diffusion models, chaining generation steps into reusable visual workflows.

> ComfyUI v0.44.0 adds ProRes video export, a dedicated VAE compute dtype control node, MiniMax-H3 model family support, and a wave of new and upgraded partner integration nodes spanning dubbing, upscaling, and music generation.

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

01 **VAE compute dtype control node** NEW 80

Adds a Set VAE Compute Dtype node to control VAE compute dtype at the node level. Usage: add the node to the graph, connect it to a VAE loader, and select the desired dtype (e.g. fp16, bf16) from the node's dropdown before running the workflow.

Override the VAE compute dtype for a workflow when you want to trade speed for precision or reduce VRAM usage.

📍 In the node graph, right-click the canvas and add the Set VAE Compute Dtype node, then connect it to your VAE loader and select the desired dtype (e.g. fp16, bf16) from the node's dropdown before running the workflow.

— Named node plus exact dropdown values and connection steps v0.44.0

02 **Upgrades to existing partner nodes** IMPROVED 65

Improves existing partner nodes: adds `GPT-6.1 Terra and Sol` model support to the `OpenAI` partner nodes, `1080p`, `2K`, and `4K` resolution targets to the BFL Flux Video Upscale V2 partner node, context window region editing to the `Ideogram 4.5 Precise Edit` partner node, and the Wonder 4 model to the Topaz Image Enhance partner node.

— Names every upgraded node and the new capability each gained v0.44.0

03 **New partner nodes for dubbing, texture, upscaling, and music** NEW 60

Adds new partner nodes: VEED Dubbing for AI-powered video dubbing, Patina texture nodes, Pruna P-Video Upscale, `MiniMax H3 Max` Recast, Extend, Reframe, Insert, Lip Sync, and Camera Control nodes, and `Sonilo Music 1.5`.

— Enumerates every new partner node name but no usage detail v0.44.0

THINNER COVERAGE BELOW

04 **MiniMax-H3 model family and new generation nodes** NEW

58

Adds support for the `MiniMax-H3` latent upscaler model and `MiniMax Sol-H3-VAE`, context windows for `MiniMax-H3`, the `PH3Image` model (`CORE-697`), and a `TextGenerateV2` node with autogrow image inputs.

— Names every model and node but gives no usage instructions v0.44.0

05 **ProRes video saving and chroma subsampling** NEW

45

The video save node now supports saving videos in ProRes format, with an option to choose chroma subsampling for exported ProRes files.

— Names the format and option but not the node or UI path v0.44.0

06 **`execution_meta` field in job detail API** NEW

40

Adds `execution_meta` to the job detail API response.

— Named API field but no schema or example response v0.44.0

07 **WaveSpeed nodes moved to Comfy Router** IMPROVED

30

Moves WaveSpeed FlashVSR and image upscale nodes to Comfy Router.

— States the move but no migration detail or rationale v0.44.0

WAS THIS USEFUL? 👍 👎

llamastash

Feedback ⓘ v0.11.0 NOTES ↗ RANK

Zero-overhead, terminal-native local-LLM orchestrator. Run any engine, any model via the same proxy.

> llamastash's v0.11.0 adds model-removal and benchmarking CLI verbs, a configurable unified-memory admission reserve with a live TUI editor, a hardware-ranked HuggingFace pull dialog, and a breaking change to the structured JSON output of `init` / `recommend`.

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

01 Configurable unified-memory admission reserve NEW

91

Set `daemon.unified_memory_reserve_percent` in `config.yaml` to `auto` or an integer from 0 to 99 to control how much memory admission holds back on unified-memory hosts (Apple Silicon, AMD/Intel integrated GPUs, NVIDIA coherent UMA). Pressing `m` in the TUI opens an editor with a live preview of the available memory pool, reserve, and resulting admission budget; saving writes to `config.yaml` and applies without a daemon restart.

Cap the admission budget on a 40 GiB Apple Silicon host to leave headroom for other GPU workloads.

```yaml
daemon:
  unified_memory_reserve_percent: 10
```

— Config key, value range, TUI key, and apply behaviour specified v0.11.0

02 New `delete` and `bench` CLI verbs NEW

90

`llamastash delete <model>` removes a model from the CLI, supporting `--yes` to skip the confirmation prompt and `--json` for machine-readable output, and exits `75` on refusal or failure. `llamastash bench <model>` benchmarks throughput, accepting `--preset`, `--runs`, `--ctx`, and `--json` flags to compare median token/s and time-to-first-token across presets before committing a context-size change.

Remove a model non-interactively in a script or agent pipeline, with machine-readable confirmation of what was deleted.

```
$ llamastash delete qwen --yes --json
```

Compare throughput across two presets before committing a context-size change to config — median token/s and TTFT per preset.

```
$ llamastash bench qwen --preset small,big --runs 5 --ctx 4096 --json
```

— Exact commands, flags, and exit code all given `v0.11.0`

---

03 **`facts` object replaces `justification` in JSON output**

85

`BREAKING`

Running `init --json` or `recommend --json` now returns each recommendation's numbers as a `facts` object with fields `params`, `quant`, `peak_bytes`, `host_total`, `bench`, and `spill_percent`, replacing the previous pre-rendered `justification` string. This is a breaking change: scripts reading `justification` from `init --json` or `recommend --json` output will break.

— Exact commands, old and new fields, and breakage named `v0.11.0`

THINNER COVERAGE BELOW

04 **`mode_hint` field for stt/image/tts launches** `IMPROVED`

55

Launching an `stt`, `image`, or `tts` entry now shows that mode in the `status` table, exposed as the `mode_hint` field in `--json` output.

— Names field and command but lacks a runnable example `v0.11.0`

05 **Hardware-ranked HuggingFace pull dialog** `IMPROVED`

50

The HuggingFace pull dialog now opens ranked by hardware compatibility — the same list `init` and `llamastash recommend` show — displaying the recommended quant, a fit mark, and an estimated size for each entry.

— Describes behaviour but gives no exact UI navigation `v0.11.0`

└──▷ **BREAKING ON UPGRADE**

! Scripts that read the `justification` field from `init --json` or `recommend --json` output will break; that field is replaced by a `facts` object containing `params`, `quant`, `peak_bytes`, `host_total`, `bench`, and `spill_percent`.

WAS THIS USEFUL? 👍 👎

llmfit

Feedback ⓘ v1.1.19 NOTES ↗ CODE ↗ RANK

Hundreds of models & providers. One command to find what runs on your hardware.

> llmfit's biggest addition this window is a preview cluster mode for finding models that fit Ray or multi-node GPU clusters, alongside a refinement to how PCIe benchmark matches are reported.

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

01 **Cluster mode for Ray and multi-node GPU fit** NEW 83

A new `--cluster` flag (preview) finds models that fit a Ray or multi-node GPU cluster; `--profile` is rejected when `--cluster` is set, and `--force-runtime` is honored per-node. Cluster discovery sends the Ray auth token and warns if connecting over plain HTTP instead of HTTPS, and `cluster.toml` lets you version cluster configuration for reproducible multi-node setups.

— Names flag, config file, and per-node behavior but marked preview. v1.1.19

THINNER COVERAGE BELOW

02 **Exact vs legacy PCIe benchmark match labeling** IMPROVED 30

llmfit now distinguishes exact PCIe benchmark matches from legacy calibrated ones in fit results, so users can see whether a result came from a precise benchmark or an older calibrated estimate.

— Describes the change but no concrete surface or example shown. v1.1.19

WAS THIS USEFUL? 👍 👎

Open WebUI

Feedback ⓘ v0.13.0 NOTES ↗ RANK

Open WebUI is a self-hosted chat interface for local and remote models, with RAG over uploaded documents, tool calling, and multi-user access control.

> Open WebUI's v0.13.0 release introduces a tabbed chat workspace with Desktop and Browser tabs for computer-use and browser-automation workflows, alongside a Git panel, side chats, comment threads, MCP app rendering, multi-model sub-agents, and new tool-creation and knowledge-base controls.

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

01 Tabbed workspace with Desktop and Browser tabs NEW

75

A tabbed workspace accessible from the Tabs button at the top of a chat supports files, multiple terminal sessions, artifacts, previews, voice calls and comments as reorderable tabs. It includes a Desktop tab for viewing and controlling a connected Open Terminal's desktop with mouse, keyboard, clipboard and audio support for computer-use workflows, a Browser tab for opening websites and local dev servers beside a chat (with Chrome mode on a compatible Open Terminal connection providing a shared browser session for automation), and a full-width layout option for larger screens that keeps the conversation as its own tab, with the layout choice remembered between sessions.

— UI path given but no config key or command v0.13.0

02 Knowledge base file count and size limits NEW

75

Knowledge base file counts and sizes can be capped with the `KNOWLEDGE_MAX_FILE_COUNT`, `KNOWLEDGE_MAX_FILE_SIZE` and `KNOWLEDGE_MAX_UPLOAD_COUNT` environment variables, or under Admin Settings > Documents; per-knowledge-base limits are also settable by the owner under Advanced at creation time.

Restrict a shared deployment so each knowledge base can hold at most 500 files, each no larger than 50 MB, with at most 20 files per upload.

```ini
KNOWLEDGE_MAX_FILE_COUNT=500
KNOWLEDGE_MAX_FILE_SIZE=52428800
KNOWLEDGE_MAX_UPLOAD_COUNT=20
```

— Three env vars named plus a concrete runnable example v0.13.0

03 Git panel in the side panel NEW

70

A Git panel in the side panel lets you review diffs, stage, discard and commit changes, manage branches and worktrees, push and pull, browse history, and work with GitHub pull requests when the terminal's folder is a Git repository.

— Lists mechanisms but only a UI entry point, no commands `v0.13.0`

04 **Comment threads across chats, notes and workspace items** NEW

70

Comments can be enabled on chats, notes, models, knowledge bases, prompts, skills and tools — including inline passage highlights and reply threads — by turning on Comments in Admin Settings > General or setting the `ENABLE_COMMENTS` environment variable.

Let users leave comment threads and mentions on chats and notes while keeping the notification inbox active on a self-hosted instance.

```ini
ENABLE_COMMENTS=true
ENABLE_NOTIFICATIONS=true
```

— Named env var and admin path, with config example `v0.13.0`

05 **Side chats via /side and /btw** NEW

65

Side chats can be triggered by typing `/side` or `/btw` in a chat or choosing Side chat in the side panel; access is controlled by the new Allow Side Chat permission, which is off by default.

Open a quick side chat to explore an alternative approach without polluting the main conversation thread.

> 📍 In any active chat, type `/side` and press Enter — a temporary side conversation opens using the current chat as context. Close its tab when done; nothing is added to the main chat.

— Named trigger commands and permission, with example `v0.13.0`

06 **Tool creation and management in Workspace** NEW

60

Models with native function calling can create a private Python tool in your workspace from chat, toggleable per model under Tool Creation in the model editor's built-in tools; separately, Workspace > Tools gains an on/off switch so a tool switched off can no longer be picked or run in any chat.

— Names both UI surfaces and toggle behaviour `v0.13.0`

THINNER COVERAGE BELOW

07 **Notification inbox for mentions and reminders** NEW

55

A notification inbox collects comment mentions, channel replies, failed automation runs and calendar reminders; administrators can disable it via Admin Settings > General or the `ENABLE_NOTIFICATIONS` environment variable.

Let users leave comment threads and mentions on chats and notes while keeping the notification inbox active on a self-hosted instance.

```ini
ENABLE_COMMENTS=true
ENABLE_NOTIFICATIONS=true
```

— Named env var but only a list of notification sources v0.13.0

08 **MCP Apps rendered in chat** NEW

50

MCP Apps render interactive apps from MCP servers in a sandboxed frame after a tool runs, with per-call confirmation required for any tool call the app initiates.

— Describes mechanism but no entry point or config v0.13.0

09 **Multi-model sub-agents in model editor** NEW

50

The model editor now supports multi-model sub-agent assignment, letting you assign research, coding and review to different models rather than always delegating to the same model as the parent.

— Names UI location but no config keys or limits v0.13.0

10 **Live updates for Git panel, ports and processes** IMPROVED

45

Live terminal updates now drive the Git panel, open ports list and running processes on terminals that support it, replacing the previous polling interval.

— States the change but no config or surface name v0.13.0

WAS THIS USEFUL? 👍 👎

◆ Inference Providers & GPU Clouds

Eigen Labs Darkbloom

Feedback ⓘ v0.10.7 NOTES ↗

CODE ↗ RANK

Darkbloom is a for the innovators pushing the frontier.

> Darkbloom's v0.10.7 release adds a model allow-list for coordinators, automatic coordinated updates for paired Macs, a rollback command, tensor-parallel serving for two new models, and a richer sandbox sizing plan display.

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

01 Model allow-list via enabled_models config

BREAKING

90

Set `enabled_models` in provider config to restrict which models a coordinator can load: `load_model` or `prefetch_model` requests outside the list are now refused with `model_not_selected`, and the provider no longer automatically switches to a new model build — use `darkbloom switch` instead.

— Names config key, refusal code, and required replacement command v0.10.7

02 Automatic coordinated updates for paired Macs

BREAKING

80

From v0.10.7 onward, both Macs in a two-Mac cluster update together automatically without manual intervention. A pair on v0.10.6 cannot self-update to v0.10.7 — each Mac must be updated manually with `darkbloom update`, then logged out and back in (or restarted) on both before the automatic coordinated-update feature takes effect. A pair whose Macs run different builds refuses to pair and fails closed until both are on the same release.

— Gives exact migration command and failure conditions v0.10.7

03 Update rollback command

NEW

80

`darkbloom update --rollback` rolls back to the previous release on a standalone (non-paired) Mac after a bad update. It refuses to run on a Mac that is actively serving in a pair — the cluster service must be stopped first.

Roll back to the previous release after a bad update, on a standalone (non-paired) Mac.

```
$ darkbloom update --rollback
```

— Runnable command plus clear precondition v0.10.7

04 **Sandbox setup plan display from install.sh** IMPROVED

75

`darkbloom sandbox setup` now downloads and runs `install.sh --plan --json-progress` to display the add-on's own CPU, memory, and disk sizing plan, replacing a local copy of the sizing rules.

— Names exact command and flags used v0.10.7

THINNER COVERAGE BELOW

05 **Tensor-parallel serving for gemma-4-26b-qat-4bit and mimo-v2.6-flash-mopd**

NEW

58

Tensor-parallel serving for `gemma-4-26b-qat-4bit` and `mimo-v2.6-flash-mopd` is now active in this build, declared in `ClusterPublicModels` alongside GLM-5.3-Flash and Qwen3.8-27B.

— Names models and config list but no usage steps v0.10.7

└──▷ **BREAKING ON UPGRADE**

! Both Macs in a pair must run v0.10.7: a pair whose Macs run different builds refuses to pair and fails closed until both are on the same release.

! A pair on v0.10.6 cannot self-update to v0.10.7 — run `darkbloom update` on each Mac manually, then log out and back in (or restart) on both before the automatic coordinated-update feature takes effect.

! With `enabled_models` set, a coordinator-initiated `load_model` or `prefetch_model` for a model outside the list is refused with `model_not_selected`; the provider no longer auto-switches to a new model build — `darkbloom switch` is now required.

! `darkbloom update --rollback` refuses to run on a Mac that is actively serving in a pair — the cluster service must be stopped first.

WAS THIS USEFUL? 👍 👎

Fireworks AI

Feedback ⓘ New Serverless mode: Ultra-fast (Preview)

NOTES ↗ RANK

Fireworks AI provides hosted inference, model fine-tuning, and deployment APIs for open-weight models.

> Fireworks AI introduced an Ultra-fast Serverless inference mode in preview, offering a major speed boost for latency-sensitive and agentic coding workloads.

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

01 Ultra-fast Serverless mode for DeepSeek V4.1 Flash NEW

83

A new preview Serverless inference mode targets 6–8x generation speed, and up to 10x for agentic coding tasks. Access it by setting `model` to `accounts/fireworks/routers/deepseek-v4p1-flash-1017-ultrafast` when calling the chat completions API.

Call the Ultra-fast variant of DeepSeek V4.1 Flash 1017 when you need minimum latency in an agentic coding pipeline.

```
$ curl -X POST
https://api.fireworks.ai/inference/v1/chat/completions \
  -H 'Authorization: Bearer <YOUR_API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "accounts/fireworks/routers/deepseek-v4p1-flash-
1017-ultrafast", "messages": [{"role": "user", "content": "
<prompt>"}]}'
```

— Names exact model router string and speed multipliers with runnable curl example

New Serverless mode: Ultra-fast (Previe…

WAS THIS USEFUL? 👍 👎

Anyscale

Feedback ⓘ Version 0.26.113 release notes NOTES ↗ RANK

Anyscale provides a managed Ray platform for distributed AI development, training, batch processing, and model serving.

> Anyscale's 0.26.113 release adds workspace deletion, multi-cloud traffic splitting for services, a SCIM safety override, and a more automated KubeRay cloud setup flow.

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

01 KubeRay cloud setup detection and automation IMPROVED

90

`anyscale cloud setup --stack kuberay` now checks for a running KubeRay operator and offers to install the bundled one, controllable with `--kuberay-operator install` to always install it or `--kuberay-operator skip` to manage it yourself (default `auto`). It also detects Nvidia DCGM exporters for GPU metrics, with `--dcgm-namespace` to disambiguate multiple exporters and `--ingress-class` to name the ingress class to use, and offers to run a sample RayJob after setup, controllable with `--test-job` to run it automatically or `--no-test-job` to skip it.

Set up an Anyscale KubeRay Connect cloud, always installing the bundled KubeRay operator, targeting a specific ingress class, and running a post-setup test job automatically.

```
$ anyscale cloud setup --stack kuberay --kuberay-operator install
--ingress-class traefik --test-job
```

— All flags named verbatim with a runnable example command.

Version 0.26.113 release notes

02 Workspace deletion via CLI and SDK NEW

75

Delete a terminated or failed workspace with `anyscale workspace_v2 delete` or anyscale.workspace.delete(), identifying it by `--id` or by `--name` together with `--cloud` and `--project`; pass `--yes` to skip the confirmation prompt.

Delete a workspace that has terminated, skipping the interactive confirmation prompt in a CI pipeline.

```
$ anyscale workspace_v2 delete --name my-workspace --cloud my-
cloud --project my-project --yes
```

— Exact command, flags and CI-style example given. Version 0.26.113 release notes

03 **Multi-cloud traffic policy for services** NEW 75

Services can split traffic across multiple clouds by adding `traffic_policy` to the service config, supporting `EQUAL` (even split) and `STATIC` (weight-proportional) policies. The traffic policy of a running service can be changed without redeploying it using `anyscale service update-traffic-policy` or anyscale.service.update_traffic_policy().

— Names config key and command but no worked example. Version 0.26.113 release notes

04 **SCIM group enforcement removal override** NEW 65

Pass `--acknowledge-user-removal-count` to `anyscale scim enforce-groups` to confirm removal of more users than the safe limit when SCIM enforcement would otherwise be blocked.

— Exact flag and subcommand named, no example shown. Version 0.26.113 release notes

WAS THIS USEFUL? 👍 👎

Replicate

Feedback ⓘ The Ultimate Guide to Kling 4.0 (2026-10-27)

BLOG ↗ RANK

Replicate is a run open-source machine learning models with a cloud API

> Replicate added support for Kling 4.0, a video generation model capable of producing up to 30 seconds of 1080p footage from a single prompt.

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

01 **Kling 4.0 video generation support** NEW 55

Replicate now supports Kling 4.0, which generates up to 30 seconds of 1080p video from a text prompt and threads one continuous shot through up to 10 keyframes.

— Names model and specs but no API call or endpoint shown

The Ultimate Guide to Kling 4.0 (2026-1…

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

matrixone

Feedback ⓘ v4.2.8 NOTES ↗ RANK

AI-native HTAP database with Git-for-Data and built-in vector search, serving as the data and memory backbone for intelligent agents and applications.

> MatrixOne v4.2.8 adds finer control over IVF vector search queries, letting users override probe counts per query and tightening validation of rank options.

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

01 **Query-level IVF nprobe tuning** IMPROVED 50

Individual IVFFLAT and IVFPQ queries can now override the session-level probe count via query-level IVF nprobe tuning, rather than relying solely on the session setting. Additionally, invalid or unknown rank options now return errors instead of being silently ignored.

— Names the mechanism but no exact flag/syntax given v4.2.8

WAS THIS USEFUL? 👍 👎

stashbase

Feedback ⓘ v2.18.9 NOTES ↗ RANK

stashbase is an open-source, local-first writing workspace where you write with Claude Code and Codex, drawing on your own sources.

> stashbase's v2.18.9 release focuses on deeper diagnostics for failed local writes, adding richer evidence capture, clearer timeline distinctions, and safer handling of restricted attachments.

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

01 **Diagnostic evidence capture for failed writes** IMPROVED

48

Stashbase now retains path, project, session, edit-match, version, and filesystem-stage evidence for failed host text writes under the existing diagnostics switch. Timelines distinguish a later success on the same target from a success at a different target, including unchanged writes, and full paths are now preserved in restricted local attachments, with routine events carrying correlation IDs and classifications.

— Describes mechanism but no exact flag or config name given. v2.18.9

WAS THIS USEFUL? 👍 👎

infino

Feedback ⓘ v0.11.21 NOTES ↗ RANK

Embedded retrieval library built on Parquet. Fast, efficient, and scalable.

> Infino adds a sorted range index for numeric column predicates and significantly speeds up large-scale ingest through parallelized term index encoding.

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

01 **Faster 1B-row ingest via parallel term index tiering** IMPROVED

60

Supertable 1B-row ingest is now 2× faster, dropping from 381s to 191s with query latency unchanged. The term index now encodes its tiers on a dedicated thread, cutting ingest time by an additional 15% (0.85× of previous).

— Concrete before/after numbers but no actionable config or flag v0.11.21

THINNER COVERAGE BELOW

02 **Sorted range index for numeric predicates** NEW

40

Infino adds a per-superfile sorted range index that accelerates range predicates on numeric columns.

— Names the mechanism but no config or API surface given v0.11.21

WAS THIS USEFUL? 👍 👎

aegis-stack

Feedback ⓘ v0.18.0 NOTES ↗ RANK

A production-ready FastAPI platform with modular components and a built-in control plane.

> v0.18.0 adds a configurable meeting bot, full-transcript meeting search with a reshaped retrieval API, CI-enforced architectural pattern checks, and rolling deploys pushed from GitHub Actions, alongside breaking renames of memory modules to context modules and consolidated meeting agent tools.

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

01 Meeting search and full-transcript retrieval API

BREAKING

95

Search meetings with `meetings search` and browse series with `meetings series` from the CLI, with search matching any supplied words and ranking by best match first. `GET /api/v1/meetings/{id}` now retrieves a full meeting — its notes and every transcript line — or a whole series, with the response nested under the `meeting` key (previously a flat response), and the transcript tab copies the call as plain lines. The `meetings_search` and `meetings_series` agent tools are replaced by a single `meetings` tool; agents granted either old tool keep the grant but it now does nothing — grant `meetings` instead.

Fetch a meeting's full transcript and notes in one call after the v0.18.0 response-shape change.

```
$ curl -s https://your-host/api/v1/meetings/{id} | jq '.meeting'
```

— Names endpoint, response-key change and CLI commands precisely v0.18.0

02 Architectural pattern checks in CI

NEW

90

Run `patterns check --strict` in CI on changed files; set `ILLIANA_BLOCKS=true` to fail the job on violations, or leave it unset to comment only. A new `parts.own-facade` rule flags any service reaching past its own facade from its router, tools, jobs, CLI, or pages.

Gate a PR on architectural violations — comment on every violation but only block merge when you explicitly opt in.

```
$ ILLIANA_BLOCKS=true patterns check --strict
```

— Flag, env var and rule name all given with a runnable command v0.18.0

03 **Rolling deploys and env push via GitHub Actions** NEW

90

Push your production environment to GitHub Actions with `aegis deploy-cd-env`, which writes `.env.deploy` as a secret for use in deploy workflows. Re-initialise existing deploy workflows to pick up rolling deploys and env push support with `aegis deploy-cd-setup --force`.

Push your production .env to GitHub Actions as a secret before triggering a rolling deploy from CI.

```
$ aegis deploy-cd-env
```

— Exact CLI commands and generated secret file named v0.18.0

04 **Memory modules renamed to context modules** BREAKING

75

Memory modules are renamed to context modules: the database table, the agent column, the CLI group, and the API paths all use the new name, requiring the agents migration to move rows (ids preserved, saved memory keeps its old name). Context modules now hold per-subject text so each conversational turn renders only the text for the subject the app passes in, loaded in one query.

— Clear migration note but API paths not named verbatim v0.18.0

05 **Meeting bot with voice-controlled attention** NEW

60

Adds a meeting bot that answers only the meeting owner by default, stops on voice commands (stop , be quiet , never mind), resumes on go on , and wakes within a fraction of a second of a caption naming it.

— Behavior well described but no flag or config to invoke it v0.18.0

THINNER COVERAGE BELOW

06 **User-scoped live voice chat with budget enforcement** NEW

40

Live calls now run as the signed-in user and are refused when the budget is spent, enabling in-app voice chat scoped to each user's context.

— Describes behavior only, no named surface or setting v0.18.0

└──▷ ALSO FROM THESE RELEASES

.
.
.

└──▷ BREAKING ON UPGRADE

! Memory modules are renamed to context modules: the database table, the agent column, the CLI group, and the API paths all use the new name. Run the agents migration to move rows (ids are preserved). Saved memory keeps its name.

! The `meetings_search` and `meetings_series` tools are replaced by a single `meetings` tool. Agents granted either old tool keep the grant but it does nothing; grant `meetings` instead.

! `GET /api/v1/meetings/{id}` now returns the full meeting (notes and every line); the bare meeting row is now nested under the `meeting` key, breaking any code that reads the old flat response.

WAS THIS USEFUL?  

synthadoc

Feedback ⓘ v1.5.1 NOTES ↗ RANK

Synthadoc: An open-source LLM knowledge compilation engine that turns raw documents into structured, local-first wikis. A transparent, human-readable alternative to traditional RAG, which can be self-managed and self-improved without the use of any tools.

> Synthadoc's v1.5.1 release adds a per-block approval workflow with conflict and sentence-loss safeguards for resolver edits, a CodeMirror 6 fallback block editor, broader startup provider checks, and a small bump in hybrid search recall.

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

01 **Per-block approval for resolver edits** NEW 85

Synthadoc's contradiction resolver and content-health resolver now express changes as `[old_text, new_text]` block pairs, letting you approve or skip each edit independently in the web UI's `ScopedConfirmCard` before anything is written. An overlap conflict detector rejects both blocks with `"conflict"` when two proposed edits target overlapping raw spans, and a sentence-loss guard rejects any edit that drops more than 20% of the sentences in `old_text`, labelling the block `"sentence_loss"`. An optional `rationale` field, populated by the resolver when the reason for a change isn't obvious, now appears above each block's card header.

— Names fields, labels and UI component; mechanism fully explained v1.5.1

02 **Hybrid search fusion tuning** IMPROVED 70

Hybrid search fusion now pins only the single best BM25 hit to the front of the fused candidate list, improving paraphrase recall@5 on a 1,985-page Wikipedia corpus from 68% to 70%.

— Concrete before/after metric but no user-facing action v1.5.1

03 **CodeMirror 6 fallback block editor** NEW 65

The block editor falls back to a CodeMirror 6 editor with an Edit/Preview toggle and an inline markdown linter covering unclosed fenced code blocks, unclosed wikilinks, and heading level skips, plus a collapsible Reference panel showing the page before any changes.

— Describes editor and linter scope, but only a UI path to act on v1.5.1

04 **Startup checks across all 8 providers** IMPROVED 60

Synthadoc now checks all 8 providers' API keys at startup instead of 4, and raises a config validation error when every provider section in `config.toml` is commented out.

— Names config file and exact provider count change v1.5.1

THINNER COVERAGE BELOW

05 **New config validation error codes** NEW 35

Three new config error codes — `ERR-CFG-003`, `ERR-CFG-004`, `ERR-CFG-005` — are now documented in the design doc error table.

— Bare naming of codes with no behavioural context v1.5.1

WAS THIS USEFUL? 👍 👎

piragi

Feedback ⓘ v1.3.0 NOTES ↗ RANK

> piragi v1.3.0 introduces reasoning-intensive retrieval, adding a deliberate reasoning step before retrieval and two new reranker types for scoring candidates with LLM reasoning.

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

01 CausalLLMReranker and ListwiseReasoningReranker reranker types

NEW 75

New `CausalLLMReranker` and `ListwiseReasoningReranker` can be selected via the `reranker_type` config key, using `reranker_type="causal_llm"` to rerank with causal LLM scoring or `reranker_type="listwise"` to score candidates jointly across the full result set via listwise reasoning.

Run fully local reasoning-intensive retrieval over a document corpus using an open-weights model, without any external API calls.

python

```python
from piragi import Ragi

kb = Ragi(
    "./docs",
    config={
        "retrieval": {
            "use_think_then_retrieve": True,
            "reranker_type": "listwise"
        }
    }
)
answer = kb.ask("Which theorem applies to this boundary
condition?")
print(answer.text)
```

Use causal LLM reranking on top of hybrid search to improve precision on complex multi-step queries.

```python
from piragi import Ragi

kb = Ragi(
    "./docs",
    config={
        "retrieval": {
            "use_hybrid_search": True,
            "reranker_type": "causal_llm"
        }
    }
)
chunks = kb.retrieve("Explain the authentication flow", top_k=10)
for chunk in chunks:
    print(chunk.score, chunk.chunk[:80])
```

— Names both reranker classes, config key and values, with examples v1.3.0

---

02 **ThinkThenRetrieve reasoning step before retrieval** NEW

70

Pass `use_think_then_retrieve=True` in the `retrieval` config to chain deliberate reasoning before each retrieval step, as in Ragi("./docs", config={"retrieval": {"use_think_then_retrieve": True}}).

Run fully local reasoning-intensive retrieval over a document corpus using an open-weights model, without any external API calls.

```python
from piragi import Ragi

kb = Ragi(
    "./docs",
    config={
        "retrieval": {
            "use_think_then_retrieve": True,
            "reranker_type": "listwise"
        }
    }
)
answer = kb.ask("Which theorem applies to this boundary
condition?")
print(answer.text)
```

— Names exact config flag and runnable example v1.3.0

WAS THIS USEFUL?  

Qdrant

Feedback ⓘ announced · Nov 2, 2026 BLOG ↗ RANK

Qdrant is an open-source vector database and search engine that stores, indexes, and retrieves high-dimensional vectors for semantic search and retrieval-augmented generation.

> Qdrant 1.20 introduces TurboQuant-based quantization datatypes that shrink vector storage without a rescoring penalty, and adds per-element indexing for nested payload fields to speed up nested filter queries.

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

01 turbo8/turbo16 quantization datatypes NEW

95

Setting `datatype` to `turbo8` or `turbo16` in a `PUT /collections/{collection_name}` request stores vectors at 8-bit or 16-bit precision using TurboQuant rotation-based compression, cutting storage to one-quarter or one-half of float32 while keeping recall within ~1% of float32 and requiring no rescoring step.

Create a collection using 8-bit TurboQuant compression to cut vector storage to one-quarter of float32 without a rescoring penalty.

```
$ curl -X PUT 'http://localhost:6333/collections/my_collection' \
  -H 'Content-Type: application/json' \
  -d '{
    "vectors": {
      "size": 1536,
      "distance": "Cosine",
      "datatype": "turbo8"
    }
  }'
```

— Names exact endpoint, flag, values and gives runnable example. launch-20261102-6782fef3

02 Per-element indexing for nested payload fields NEW

90

Calling `PUT /collections/{collection_name}/index` with `nested` set to the parent array key (e.g. `'authors'`) on a nested field like `authors[].first` creates a per-element payload index, enabling index-only evaluation of nested filters that serves up to four times as many queries per second as regular nested indexes on 1 million points.

Build a per-element index on a nested array field so that nested filters (e.g. author first name AND last name in the same element) run entirely from the index with no payload reads.

```
$ curl -X PUT
'http://localhost:6333/collections/my_collection/index' \
  -H 'Content-Type: application/json' \
  -d '{
    "field_name": "authors[].first",
    "field_schema": {
      "type": "keyword",
      "nested": "authors"
    }
  }'
```

— Names endpoint, config key and quantified QPS gain with example.

launch-20261102-6782fef3

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ updated · Nov 4, 2026

NOTES ↗ RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's biggest addition is deeper LLM Gateway observability and new Bedrock IAM auth, alongside a migration-forcing change to thread automations, a deprecated Turns view, new GPT-5.6 chat models, dual-stack networking, sandbox resource history, and smaller fixes to WebSocket proxying, CSV export, tagging permissions, and API key search.

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

01 LLM Gateway trace observability and headers IMPROVED

85

Trace details now show a dedicated Gateway section with gateway outcomes, policy limits, selected models, and fallback attempts, plus a Blocked badge on errored runs blocked by the gateway. Gateway responses also include an `X-LangSmith-Gateway-Metadata` header describing routing, limits, guard activity, and upstream errors, which clients can attach to their own traces as `ls_gateway_info` metadata.

— Names header and metadata key but no endpoint or flag to invoke. snapshot-20261104

02 Turns view deprecated in thread detail DEPRECATED

70

The Turns view in thread detail is deprecated and will be removed from LangSmith Cloud on October 31, 2026, and from self-hosted in v17; use the Messages or Details view instead.

— Exact removal dates and replacement views given. snapshot-20261104

03 SmithDB-only queries required for thread automations BREAKING

65

Thread annotation-queue and webhook automations now require SmithDB queries and can add one annotation queue item per matched conversation, letting reviewers inspect the full thread instead of a single run; automations using other query types must be migrated.

— Clear migration need but no exact query syntax given. snapshot-20261104

04 IAM role auth for Bedrock in LLM Gateway NEW

60

The LLM Gateway now authenticates to Amazon Bedrock with an assumed IAM role or AWS access keys as an alternative to a static Bedrock API key, with existing Bedrock API

keys taking precedence when multiple credential types are configured.

— Describes auth options and precedence but no config key named. snapshot-20261104

05 **CSV export for colon-containing dataset keys** IMPROVED

60

Datasets whose input, output, or metadata keys contain a colon — including feedback stored under a `source_run_feedback:<key>` metadata key — can now be exported to CSV.

— Names exact metadata key pattern and export format. snapshot-20261104

THINNER COVERAGE BELOW

06 **WebSocket query strings preserved in sandboxes** IMPROVED

55

WebSocket connections through a sandbox service URL now reach the target service with their query string intact, allowing dev servers that authenticate WebSocket connections with a query parameter (such as Vite's HMR token) to connect successfully.

— Concrete example (Vite HMR) but no config or flag. snapshot-20261104

07 **Resource usage history for sandboxes** NEW

45

Sandboxes now retain recent CPU, memory, and network usage history to surface resource consumption and diagnose performance bottlenecks.

— States what's tracked but no retention window or UI path. snapshot-20261104

08 **API Keys page search filters** IMPROVED

45

The API Keys page now filters keys across the full paginated list by key, description, and workspace.

— Names filter fields and page, no exact navigation path. snapshot-20261104

09 **Tag-on-create permission for datasets, projects, prompts** NEW

40

Users with the tag-on-create permission for datasets, projects, or prompts can now see and apply tags when creating those resources.

— Names permission and resource types, no UI path given. snapshot-20261104

10 **New GPT-5.6 models in LangSmith Chat** NEW

35

LangSmith Chat now offers GPT-5.6 Sol, GPT-5.6 Terra, and GPT-5.6 Luna alongside existing OpenAI models.

— Just lists model names, no usage detail. snapshot-20261104

11 **IPv4/IPv6 dual-stack connectivity** NEW

25

LangSmith services now accept both IPv4 and IPv6 connections.

— Bare statement, no config or scope detail. `snapshot-20261104`

└──▷ BREAKING ON UPGRADE

! The Turns view in thread detail is deprecated and will be removed from LangSmith Cloud on October 31, 2026, and from self-hosted in v17; use the Messages or Details view instead.

! Thread annotation-queue and webhook automations now require SmithDB queries — automations using other query types must be migrated.

WAS THIS USEFUL? 👍 👎

MLflow

Feedback ⓘ v3.18.0 NOTES ↗ RANK

MLflow tracks, evaluates, and monitors machine learning and LLM applications, covering experiment tracking, tracing, prompt management, and a model registry.

> MLflow 3.18.0 adds GenAI agent reliability tooling—automatic issue detection, a skill registry, online evaluation, and statistical regression comparison—alongside tracing, judge, and security hardening updates, plus several breaking changes to model downloads, logged-model search, and MCP server identity handling.

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

01 **Model code execution control flag** DEPRECATED 80

Model-supplied code execution can now be refused by setting `MLFLOW_ALLOW_MODEL_CODE_EXECUTION=false`; this deprecates `MLFLOW_ALLOW_PICKLE_DESERIALIZATION` while preserving its pickle-only behavior.

— Names exact env vars and behavior, directly settable v3.18.0

02 **MCP server version identity case-sensitivity change** BREAKING 80

MCP server version identity is now case-sensitive on MySQL and SQL Server; case-mismatched references must be corrected before upgrading by running `mlflow db upgrade <database-url>` with a backup and maintenance window. On case-insensitive databases, registry and MCP permissions now use stored resource names, so differently cased grants must be re-saved.

— Gives exact migration command and affected databases v3.18.0

03 **Custom quality criteria in GenAI evaluate** NEW 75

Adds an experimental `quality_criteria` parameter to `mlflow.genai.evaluate` for specifying custom evaluation criteria beyond built-in metrics.

Add custom quality criteria to a GenAI evaluation run to go beyond built-in metrics.

```python
import mlflow

results = mlflow.genai.evaluate(
    data=eval_dataset,
    predict_fn=my_agent,
    quality_criteria=["The response must cite sources", "The response must be under 200 words"],
)
```

— Named parameter and API with a runnable code example v3.18.0

04 **Hierarchical session support in tracing** IMPROVED 70

Offline session scorers can now select a level with `session_level`, and `session_id` can be a list in `update_current_trace` and `tracing.context` to identify hierarchical sessions.

— Named parameters and functions but no usage example v3.18.0

05 **Online Evaluation for production traces** NEW 65

MLflow adds Online Evaluation to run code-based checks automatically against new traces and multi-turn conversations in production; it is experimental and requires a self-hosted server with job containers.

— States requirements but no exact setup command v3.18.0

06 **Automatic Issue Detection for agent failures** NEW 60

MLflow adds an Automatic Issue Detection feature that groups related agent failures into a dedicated Issues tab, links findings to trace evidence, and supports scheduled detection over new traffic.

— Explains mechanism and UI location but no exact navigation steps v3.18.0

07 **search_logged_models filtering and sorting change** BREAKING 60

`search_logged_models` now filters and sorts metrics by the highest step, then latest timestamp, on SQL stores; `datasets` is now applied to both filtering and sorting.

— Names function and field with before/after behavior, no migration steps v3.18.0

THINNER COVERAGE BELOW

08 **Statistical regression comparison for evaluation runs** NEW 55

MLflow adds a redesigned evaluation and trace comparison UI plus an experimental Python API for statistical comparison of evaluation runs, letting users distinguish genuine regressions from noise.

— Describes purpose; no concrete API function name given v3.18.0

09 **Model download URI metadata removal** BREAKING 55

Downloading models via `models:/` URIs no longer includes `registered_model_meta`; custom readers must use the URI or Model Registry directly.

— Names affected URI scheme and field, gives migration guidance v3.18.0

10 **Skill Registry for reusable agent skills** NEW 50

MLflow adds a Skill Registry — an experimental central catalog for sharing reusable skills across agents and projects, with version and alias selection per agent.

— Describes capability but no API or UI path given v3.18.0

11 **OpenAI Decisions tracing support** NEW 30

MLflow now traces OpenAI Decisions calls in both Python and TypeScript.

— Names languages supported but no further mechanism v3.18.0

12 **Databricks System One models in judges** NEW 30

MLflow adds support for Databricks System One models in custom and Guidelines judges.

— Names the integration but lacks usage detail v3.18.0

13 **TraceAudio in custom trace views** NEW 25

MLflow adds `TraceAudio` support to custom trace views.

— Bare name with no mechanism or usage detail v3.18.0

└──▷ BREAKING ON UPGRADE

! `search_logged_models` now filters and sorts metrics by the highest step, then latest timestamp, on SQL stores; `datasets` is now applied to both filtering and sorting.

! Downloading models via `models:/` URIs no longer includes `registered_model_meta`; custom readers must use the URI or Model Registry directly.

! MCP server version identity is now case-sensitive on MySQL and SQL Server; case-mismatched references must be corrected before upgrading. Run `mlflow db upgrade <database-url>` with a backup and maintenance window; on case-insensitive databases, registry and MCP permissions now use stored resource names and differently cased grants must be re-saved.

WAS THIS USEFUL?  

ClearML

Feedback ⓘ v2.2.0 NOTES ↗ RANK

ClearML provides experiment management, data versioning, pipeline orchestration, and GPU scheduling for machine learning workloads.

> ClearML v2.2.0 adds configurable zip compression for artifacts, a runtime override for API session retries, finer control over worker-side logging and resource monitoring, and a new DataView listing method.

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

01 Worker-level logging and resource monitoring toggles IMPROVED

81

Per-worker stdout/stderr capture and resource monitoring can now be disabled via `sdk.development.worker.log_stdout`, `sdk.development.worker.log_stderr`, and `sdk.development.worker.resource_monitoring` config options; additionally, `CLEARML_LOG_LEVEL` now accepts logging level names (not just numeric levels).

Suppress stdout capture and resource monitoring for a lightweight worker that does not need them.

hocon

```
sdk {
  development {
    worker {
      log_stdout: false
      log_stderr: false
      resource_monitoring: false
    }
  }
}
```

— Names all three config keys, the file, and the env var change v2.2.0

02 Configurable zip compression for stored artifacts NEW

76

Dataset chunks, folder artifacts, and offline session packages can now use a chosen compression algorithm by setting `sdk.storage.zip_compression` to `stored`, `deflated`, `bzip2`, or `lzma` in the SDK config (e.g. `~/clearml.conf`).

Use lzma compression to minimise artifact size when uploading large dataset chunks.

```hocon
sdk {
  storage {
    zip_compression: lzma
  }
}
```

— Names exact config key, file, and all four supported values v2.2.0

03 **DataView.to_list() method** NEW 62

A new DataView.to_list() method retrieves all entries in a DataView as a Python list.

Iterate over every entry in a DataView to build a custom index without loading a separate iterator.

```python
from clearml import DataView

dv = DataView()
entries = dv.to_list()
for entry in entries:
    print(entry)
```

— Names exact method with a runnable usage example v2.2.0

THINNER COVERAGE BELOW

04 **Runtime override for API session retry settings** NEW 55

API session retry behaviour can now be overridden at runtime via the `CLEARML_API_RETRY_VALUES` environment variable, which also supports vault-based configuration.

— Names env var but gives no concrete value format or example v2.2.0

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.139 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik's single release in this window adds comment-based filtering for traces and threads, plus in-panel thread search.

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

01 **Comment-based filtering for traces and threads** NEW

45

Traces and threads can now be filtered by comments, with support added across the SDK, backend, and UI.

— Names SDK/backend/UI surfaces but no exact API or flag 2.2.139

02 **Thread search in side panel** NEW

40

Users can now search within a thread directly in the thread side panel.

— Clear UI location but no further mechanism detail 2.2.139

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v3.227.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse's AI Gateway gained per-user/API-key analytics filtering, gateway key management (sorting, last-used timestamps, `AI_GATEWAY` invites), HTTP/2 support, GitHub Copilot CLI client support, and per-organization rate limits, alongside a new admin API for transferring projects between organizations, expanded audit logging, opt-in trace export field groups, and bundled skills exposed via the MCP server.

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

01 Gateway keys page sorting, last-used timestamps and AI_GATEWAY invites

NEW 75

The gateway keys page now supports sorting keys by created date, name, or expiration, shows last-used timestamps for auditing key activity, and lets admins invite team members directly with a new `AI_GATEWAY` organization membership role.

Invite a contractor who only needs AI Gateway access without granting broader org permissions.

> 📍 In the gateway keys page, open Invite team member and assign the `AI_GATEWAY` role before sending the invite.

— Names exact role and three concrete page additions v4.73.0

02 AI Gateway analytics filtering by user and API key NEW

70

AI Gateway cost and usage analytics can now be filtered by user and API key, narrowing the dashboard to a specific caller via the User and API key filter controls in the analytics view.

Narrow AI Gateway cost and usage analytics to a single API key to investigate unexpected spend.

> 📍 In the AI Gateway analytics view, use the User and API key filter controls to scope the dashboard to a specific caller.

— Names exact UI filter controls and workflow v4.73.0

THINNER COVERAGE BELOW

03 **Trace field groups in legacy traces export** NEW 50

The legacy traces export now supports opt-in trace field groups, letting you filter which fields are included via the blob-export configuration.

— Config surface named but not the exact key v3.227.0

04 **Audit logging for support access and read-access routes** IMPROVED 45

Audit logs now always record support access reads and mark them in ClickHouse; a new read-access audit gate additionally records two routes in audit logs.

— Mechanism named but routes left unspecified v4.74.0 v4.73.0

05 **Skills exposed via the MCP server** NEW 45

Bundled Langfuse skills are now exposed via the MCP server, with skill management enabled by default and version automations support for skills.

— Describes what changed but no usage steps v4.73.0

06 **HTTP/2 support for AI Gateway connections** IMPROVED 40

AI Gateway web connections now support HTTP/2, with one immediate retry on failed requests.

— Clear behaviour change, no user-facing action v4.73.0

07 **GitHub Copilot CLI and app as AI Gateway client** NEW 40

Added setup instructions for GitHub Copilot CLI and app as a supported AI Gateway client.

— Named client but no setup detail included v4.73.0

08 **HIPAA notice for AI Gateway credentials** NEW 40

The UI now shows a HIPAA notice when adding or editing AI Gateway provider credentials.

— Named UI surface, minimal further detail v4.73.0

09 **Admin API to transfer projects between organizations** NEW 35

A new admin API endpoint lets administrators transfer projects between organizations.

— No exact endpoint path given in source v4.74.0 v3.226.0

10 **Monitor-triggered automation dialog** NEW 35

A new monitor automation dialog lets users create monitor-triggered automations.

— Bare description of a new UI dialog v4.73.0

11 **Per-organization AI Gateway rate limits** NEW 25

Langfuse adds per-organization rate limits for the AI Gateway.

— Mentioned only in summary, no mechanism given v4.74.0

WAS THIS USEFUL? 👍 👎

overmind

Feedback ⓘ v0.1.26 NOTES ↗ RANK

The platform for continuously improving AI agents.

> Overmind added vLLM sleep routes for dedicated snapshot engines and support for referencing IDs returned by dedicated MCP tools.

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

01 **Accept IDs from dedicated MCP tools** IMPROVED 35

Overmind now accepts the IDs that dedicated MCP tools return, enabling downstream steps to reference tool outputs.

– Explains purpose but no concrete command or field name v0.1.26

02 **vLLM sleep routes for snapshot engines** NEW 25

Overmind adds vLLM sleep routes for dedicated snapshot engines.

– Only named in passing, no mechanism or usage given v0.1.26

UPDATED INTEGRATION Langfuse version support

> v0.1.25 NOTES ↗ STABLE
>
> Langfuse imports are now resumable with sync progress tracking; sessions are preserved and onboarding and dataset recovery are simplified.
>
> └──▷ **GET THIS VERSION** ▾
>
> ```
> $ git clone --branch v0.1.25 https://github.com/overmind-core/overmind.git
> # already have the repo? check out this version:
> $ git checkout v0.1.25
> ```
>
> Langfuse imports are now resumable, with measured sync progress so long-running imports can be continued after interruption.
>
> Overmind now supports every Langfuse version when importing traces, removing prior version restrictions.

WAS THIS USEFUL? 👍 👎

traccia-py

Feedback ⓘ v0.1.34 NOTES ↗ RANK

OpenTelemetry-native SDK for AI agent observability, tracing, evaluation, debugging, governance, and runtime policy enforcement. Framework-agnostic and built for OpenAI Agents, LangGraph, CrewAI, and LLM applications in production.

> traccia-py v0.1.34 changes the default agent-call behavior during Traccia outages to fail-closed and adds a pricing-match diagnostics function for LLM cost attribution.

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

01 policy_fail_mode controls outage behavior BREAKING

90

A new `policy_fail_mode` setting controls what happens to agent calls when Traccia is unreachable (timeout, connection error, 5xx, or invalid reply). It can be set via init(), the `@govern` decorator, or in `traccia.toml`, with `'open'` allowing calls through; policy checks now block by default instead of the previous fail-open behavior, and a missing API key or agent ID now raises `GovernanceConfigError` instead of running unchecked.

Keep agents running during a Traccia outage in a dev environment where availability matters more than strict enforcement.

```python
from traccia import init

init(api_key="...", policy_fail_mode="open")
```

Set fail-open behavior for all runs in a project without changing application code, by configuring it once in `traccia.toml`.

```toml
policy_fail_mode = "open"
```

— Names exact config key, decorator, file and values with runnable examples. v0.1.34

02 Pricing match diagnostics for LLM cost IMPROVED

65

A new traccia.pricing_matcher.match_model() function identifies how a model's pricing was resolved at runtime, and spans now record `llm.pricing.match_kind` and `llm.pricing.provider` attributes. A model name that matches no pricing key now records no cost, instead of falling back to the price of a similarly named model as before.

— Names the function and span attributes but no usage example given. v0.1.34

└──▷ BREAKING ON UPGRADE

! Policy checks now block calls by default when Traccia is unreachable (timeout, 5xx, connection error, invalid reply). Previously, the call went through. To restore the old behavior, pass init(policy_fail_mode="open") or set `policy_fail_mode = "open"` in `traccia.toml`, or choose Allow Calls under Settings → Policy Enforcement.

! A missing API key or agent ID now raises `GovernanceConfigError` instead of running unchecked.

! A model name that matches no pricing key now records no cost, instead of falling back to the price of a similarly named model.

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ announced · Nov 2, 2026 BLOG ↗

RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix now keeps its built-in LLM pricing catalog current automatically, checking for rate updates roughly once a day without requiring a server restart.

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

01 Automatic background updates for LLM pricing catalog NEW

91

Phoenix now updates its built-in LLM model pricing catalog in the background, checking roughly once a day without a server restart, so new or revised token rates are available between releases. Automatic downloads can be disabled by setting `PHOENIX_DISABLE_MODEL_COST_AUTO_UPDATE=true` on the Phoenix server, or implicitly via `PHOENIX_ALLOW_EXTERNAL_RESOURCES=false`. The active catalog's generation date is visible under Settings > Models next to Built-in prices from, and for multi-server deployments sharing a PostgreSQL database, Phoenix uses a database lock so only one server applies each update.

Prevent automatic pricing downloads in an air-gapped or change-controlled environment where pricing updates must go through a formal release process.

```
$ export PHOENIX_DISABLE_MODEL_COST_AUTO_UPDATE=true
```

— Full mechanism, exact env vars, UI path and command given. launch-20261102-07d3c10b

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ updated · Nov 4, 2026 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust made recent structured filters reusable directly from the search dropdown, alongside SQL filters.

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

01 **Structured filters in search Recent list** IMPROVED 40

The search dropdown's Recent section now shows structured filters alongside SQL filters, letting users reapply a recent structured filter with one click instead of rebuilding it.

— Explains UI behavior but no exact navigation path or config `snapshot-20261104`

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

containarium

Feedback ⓘ v0.129.0 NOTES ↗ RANK

Open-source agent runtime — SSH-native isolation, eBPF egress policy, Kubernetes + LXC backends, GPU passthrough, MCP-native CLI

> Containarium's latest releases centered on box lifecycle management — skill sets and bootstrap bundles can now be re-applied without recreating containers — alongside new run token usage tracking, spot VM preemption metrics, and an opt-in in-box Python toolchain.

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

01 Skill-set re-application via API, CLI, and MCP tool NEW

90

A box's skill sets can be re-applied at any time without recreating the container: the `apply_box_skill_sets` MCP tool is available from the platform MCP server, the new `ApplyBoxSkillSets` API is backed by a per-box skill-set manifest, proto contract, typed refusals, and client wiring, and the CLI exposes the same capability via `containarium code skill-sets`; `code install` now documents the two install paths.

Re-apply skill sets to a box after updating its manifest, without recreating the container.

```
$ containarium code skill-sets --server http://your-host:8080
```

Wire the platform MCP server into Claude Code so it can create and manage Containarium boxes, with token auth.

```json
{
  "mcpServers": {
    "containarium": {
      "command": "/usr/local/bin/mcp-server",
      "env": {
        "CONTAINARIUM_SERVER_URL": "http://your-host:8080",
        "CONTAINARIUM_JWT_TOKEN": "<your-token>"
      }
    }
  }
}
```

— Names MCP tool, API, manifest, proto contract, and exact CLI command v0.128.0

v0.129.0

02 **Run token usage tracking via `RunUsage` contract** NEW

70

Token usage across skill, planner, and graph crew runs is now tracked via the `RunUsage` contract on `BoxRun`, skill, and crew runs, and can be read from structured results returned by `ListBoxRuns` (slice 2 of the run token usage contract).

— Names the contract and API but no runnable example v0.128.0

03 **Bootstrap bundle re-apply via `code bootstrap`** NEW

65

`containarium code bootstrap` now replays a box's full bootstrap sequence on demand, re-applying its bootstrap bundle without recreating the container.

Re-apply all skill sets to an existing box after updating its skill-set definitions, without recreating the container.

```
$ containarium code bootstrap
```

— Clear mechanism and runnable command, but narrow scope v0.129.0

04 **Bootstrap-URL preflight in `code install`** NEW 65

`containarium code install --bootstrap-url` now preflights a bootstrap URL before consuming it, distinguishing spent URLs from ones that are still unconsumed.

Provision a fresh box from a bootstrap URL and confirm the URL is still unconsumed before it is spent.

```
$ containarium code install --bootstrap-url https://your-host/bootstrap/<token>
```

— Exact flag and runnable command given v0.129.0

THINNER COVERAGE BELOW

05 **Spot VM preemption metrics** NEW 45

Per-target preemption metrics for spot VM workloads can now be exported with `--watch-spot-vm`.

— Names the flag but no example or detail on output v0.128.0

06 **Opt-in Python toolchain for in-box agents** NEW 45

An opt-in Python toolchain and `agent-python` helper now provide in-box Python agent support.

— Names the helper but no usage detail given `v0.128.0`

WAS THIS USEFUL? 👍 👎

NVIDIA OpenShell

Feedback ⓘ v0.1.7 NOTES ↗ RANK

OpenShell runs autonomous AI agents in policy-controlled sandboxes with isolated filesystem, network, and credential access.

> OpenShell v0.1.7 expands OTLP gateway export with HTTP transport, auth headers, and mutual TLS, adds a custom CA bundle for sandbox egress, hashed hostnames for multi-gateway Kubernetes clusters, and replica-aware SDK routing, alongside several breaking changes including OCSF 1.8.0-only output, stricter supervisor-pool validation, and a MicroVM UID change.

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

01 OTLP gateway export over HTTP, headers, and mTLS NEW

98

Forward sandboxed agent OpenTelemetry traces to a collector without an egress rule by adding `[openshell.gateway.otlp.agent]` with `enabled = true` to `gateway.toml` (the supervisor tags each span with the sandbox ID before forwarding). Export OTLP data over HTTP by setting `protocol = "http/protobuf"` on `[openshell.gateway.otlp]` or any per-signal table, with gRPC remaining the default and an endpoint containing a query string now a configuration error. Send authentication or routing headers to OTLP collectors via `headers` or `headers_file` on `[openshell.gateway.otlp]` or a per-signal table, enabling direct MLflow ingestion of relayed agent traces via `x-mlflow-experiment-id`. Connect OTLP collectors over mutual TLS by setting `ca_file`, and optionally `cert_file` and `key_file`, on `[openshell.gateway.otlp]` or a per-signal table, with the agent trace relay and OTLP/HTTP export now also supporting `https://` collectors.

Relay sandboxed agent traces to an OpenTelemetry collector without opening an egress rule from the sandbox.

toml

```toml
[openshell.gateway.otlp.agent]
enabled = true
```

Send OTLP data over HTTP instead of gRPC and authenticate to the collector with a static header, such as an MLflow experiment ID.

toml

```toml
[openshell.gateway.otlp]
protocol = "http/protobuf"
headers = { x-mlflow-experiment-id = "42" }
```

Connect the OTLP exporter to a collector that uses a private CA and requires a client certificate.

```toml
[openshell.gateway.otlp]
ca_file = "/etc/pki/collector-ca.pem"
cert_file = "/etc/pki/gateway-client.pem"
key_file = "/etc/pki/gateway-client-key.pem"
```

— All config keys and runnable TOML snippets provided. v0.1.7

---

02 **Custom CA bundle for sandbox egress** NEW 81

Route sandbox egress through a private PKI by setting `additional_ca_bundle` under `[openshell.supervisor.network]` to a PEM bundle; it is applied on top of default roots for every first-party compute driver.

— Names exact config key and scope; lacks usage example. v0.1.7

---

03 **Native OCSF 1.8.0 records only** BREAKING 80

OpenShell now emits native OCSF 1.8.0 records only; deployments that relied on 1.1 or 1.3 records now receive 1.8.0. The `ocsf_schema_version` setting, the gateway `schema_version` fields, Helm `server.ocsfLog.schemaVersion`, and `OPENSHELL_OTLP_OCSF_SCHEMA_VERSION` are deprecated and ignored; setting them logs a warning.

— Names every deprecated setting; no migration steps beyond warning. v0.1.7

---

04 **Hashed service hostnames for multi-gateway Kubernetes** NEW 72

Give each exposed sandbox service a unique hostname in multi-gateway Kubernetes clusters by setting `service_hostname_format = "hashed"` under `[openshell.gateway]`; the default `readable` format is unchanged.

— Exact config key and value given, behaviour explained briefly. v0.1.7

---

05 **Stricter supervisor pool config validation** BREAKING 70

`config preflight` and gateway startup now reject an `[openshell.gateway.supervisor_pool]` table when the selected driver does not use shared supervision, and reject a pool recovery timeout shorter than member failure detection.

— Names the config table and validation rules clearly. v0.1.7

---

06 **MicroVM sandboxes run as image's sandbox account** BREAKING

62

On the MicroVM driver, sandboxes now run as the image's `sandbox` account instead of always running as `1000:1000`; sandboxes from images whose `sandbox` account is not UID 1000 now run under that UID.

— Explains before/after behaviour but no config to adjust it. v0.1.7

THINNER COVERAGE BELOW

07 **Replica-aware routing in SDKs** IMPROVED

55

Rust, Go, Python, and TypeScript SDKs now send exec calls straight to the gateway replica that owns the sandbox when replica routing is enabled; Go also routes port-forward and SSH tunnel calls, and TypeScript routes port-forward.

— Names SDKs and call types but no config or API surface. v0.1.7

08 **Direct TLS listener for sandbox services** NEW

45

A new opt-in supervisor TLS listener lets declared HTTP and WebSocket sandbox services accept application traffic directly, without routing through the gateway; disabled by default.

— No config key or flag named to enable it. v0.1.7

09 **Typed Python gRPC stubs** IMPROVED

30

Python gRPC bindings now ship typed sync and async stubs so editors and type checkers know the concrete request and response types.

— Bare description with no names or usage path. v0.1.7

└──▷ **BREAKING ON UPGRADE**

! OpenShell now emits native OCSF 1.8.0 records only; deployments that relied on 1.1 or 1.3 records now receive 1.8.0. The `ocsf_schema_version` setting, the gateway `schema_version` fields, Helm `server.ocsfLog.schemaVersion`, and `OPENSHELL_OTLP_OCSF_SCHEMA_VERSION` are deprecated and ignored; setting them logs a warning.

! `config preflight` and gateway startup now reject an `[openshell.gateway.supervisor_pool]` table when the selected driver does not use shared supervision, and reject a pool recovery timeout shorter than member failure detection.

! On the MicroVM driver, sandboxes now run as the image's `sandbox` account instead of always running as `1000:1000`; sandboxes from images whose `sandbox` account is not UID 1000 now run under that UID.

WAS THIS USEFUL?  

Docker Sandboxes

Feedback ⓘ v0.51.0 NOTES ↗ RANK

Docker Sandboxes runs AI agents in isolated Docker environments with controlled filesystem, network, and tool access.

> Docker Sandboxes v0.51.0 adds a batch of new CLI flags and config options for sandbox creation, networking, FUSE filesystems, secrets and MCP tooling, alongside a significant breaking overhaul of how built-in agent kits and images are published and resolved.

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

01 **New mount and run-control flags for sbx create/run** NEW 90

`sbx create` / `sbx run` gain flags to mount additional workspaces at custom paths via `PATH:TARGET[:ro]` (or `additionalWorkspaces[].target` in environment files), suppress interactive prompts including Docker sign-in with `--no-input` (which reports the command needed to continue), suppress creation progress output while keeping command results with `--no-progress`, start the agent in a specific directory with `--workdir` (including on reattach), and attach no static MCP servers via `--static-mcp none`.

— Five named flags with exact syntax, fully actionable v0.51.0

02 **Kit publishing and versioning overhaul** BREAKING 85

Built-in agent names now resolve to published kits at `docker.io/sbx/<name>:latest`, and the `kit.allowExtractedAgents` setting is removed. Sandboxes created from embedded Claude or Codex specifications must be recreated before using `sbx kit add`, `sbx run --model`, or `sbx move` (save files and unpushed commits first). The shell agent now uses a published Debian 13 v3 kit, requires registry access on first use, and needs the `shell-credentials` mixin for provider credentials; v1 and v2 kits cannot be combined with v3 kits, so older mixins must be upgraded to v3 first. The `docker/sandbox-templates` images for `codex`, `cursor-agent`, `devin`, `docker-agent`, and `opencode` (including `-docker` variants) are no longer published, and `docker/sandbox-templates:shell` / `:shell-docker` are no longer refreshed — use `docker.io/sbx/shell-image` and `docker.io/sbx/shell-docker-image` instead.

— Extensive named surfaces but migration steps are prose v0.51.0

03 **Read custom secrets from stdin** NEW 75

`sbx secret set-custom --value-stdin` reads a secret's value from standard input, keeping it out of command arguments and shell history.

Pass a secret to a custom tool without exposing its value in shell history or process listings.

```
$ echo 'my-secret-value' | sbx secret set-custom my-api-key --value-stdin
```

— Exact command plus runnable usage example v0.51.0

04 **Persistent telemetry disable setting** NEW 75

`sbx settings set telemetry.disabled true` persistently disables local CLI and daemon telemetry; combined with `sbx daemon restart` it stops usage data collection immediately.

Disable telemetry for both the CLI and the running daemon so no usage data leaves the machine.

```
$ sbx settings set telemetry.disabled true && sbx daemon restart
```

— Exact command with runnable example v0.51.0

05 **Environment file defaults for sbx env run and sandbox sizing** NEW 70

Environment files can now set `entrypoint` and `cmd` defaults for `sbx env run`, and `sandboxOptions` additionally accepts `rootSize`, `dockerSize`, and `clonedWorkspaceSize`.

— Named config keys but no worked example v0.51.0

06 **Disable bundled model server** NEW 70

Setting `model.server.enabled` to `false` and running `sbx daemon restart` disables the bundled model server and its inference-runtime download.

— Config key and restart command both named v0.51.0

07 **FUSE filesystem support in local sandboxes** NEW 65

FUSE filesystems can now be enabled in local sandboxes via the `sandbox.fuse` setting or the environment option `sandboxOptions.fuse`.

— Named setting, no further mechanism given v0.51.0

08 **Custom secrets now require destination hosts** BREAKING 65

Custom secrets without destination hosts no longer substitute credentials; existing mappings must add explicit destination hosts while preserving their placeholders and other settings.

— Clear before/after but no concrete config example v0.51.0

09 **HTTP governance policy check command** NEW 60

The new `sbx policy check http` command checks HTTP governance policy behavior.

— Named command but behavior described only briefly v0.51.0

10 **Clipboard copy requires opt-in setting** BREAKING 60

Copying text from a local sandbox to the host clipboard now requires opting in via the `clipboard.textCopy` setting.

— Named setting, no migration command given v0.51.0

THINNER COVERAGE BELOW

11 **MCP tool names prefixed by server** NEW 55

Enabling `mcp.useServerNamePrefixes` prefixes MCP tool names with their server name.

— Single named setting, minimal elaboration v0.51.0

12 **Minor CLI conveniences: version flag and port alias** NEW 55

`sbx --version` now prints the client version without starting the daemon, and `-p` is added as a short alias for `--publish` in `sbx ports`.

— Two small named flags, no further detail v0.51.0

└──▷ **BREAKING ON UPGRADE**

! Built-in agent names now resolve to published kits at `docker.io/sbx/<name>:latest`; the `kit.allowExtractedAgents` setting is removed.

! Sandboxes created from embedded Claude or Codex specifications must be recreated before using `sbx kit add`, `sbx run --model`, or `sbx move` — save files and unpushed commits before removing.

! The shell agent now uses a published Debian 13 v3 kit and requires registry access on first use; add the `shell-credentials` mixin to provide provider credentials.

! V1 and v2 kits cannot be combined with v3 kits — upgrade older mixins to v3 before combining them with the new built-in kits.

! The `docker/sandbox-templates` images for `codex`, `cursor-agent`, `devin`, `docker-agent`, and `opencode` (including `-docker` variants) are no longer published.

! The `docker/sandbox-templates:shell` and `:shell-docker` images are no longer refreshed — use `docker.io/sbx/shell-image` and `docker.io/sbx/shell-docker-image` for custom workloads.

! Copying text from a local sandbox to the host clipboard now requires opting in with the `clipboard.textCopy` setting.

! Custom secrets without destination hosts no longer substitute credentials — add explicit destination hosts to existing mappings while preserving their placeholders and other settings.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ updated · Nov 4, 2026 NOTES ↗ RANK

The Claude Developer Platform is the API for building applications on Anthropic's Claude models that generate, analyze, and automate work.

> Anthropic added text watermarking to AI-generated output across its latest Claude models, with a notable gap for customized sampling parameters.

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

01 **Text watermarking for Claude model output** NEW 57

Anthropic now embeds a text watermark in content generated by Claude Mythos 5, Claude Opus 4.7, and Claude Sonnet 4.6 to mark AI-generated output. On Claude Sonnet 4.6, requests that set `temperature`, `top_p`, or `top_k` to a non-default value may not carry the watermark.

— Names models and parameters but no API detail or verification method. snapshot-20261104

WAS THIS USEFUL? 👍 👎

stealth Crouton Alpha

Feedback ⓘ released · Nov 4, 2026

MODEL ↗

Crouton Alpha from stealth was released on 3 Nov 2026. 1049K-token context.

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

Tencent Hy4 preview

Feedback ⓘ new snapshot · Nov 4, 2026

MODEL ↗

Hy4 preview is a chat model from Tencent.

New snapshot 20261104 listed on OpenRouter (tencent/hy4), first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

Supersonic Labs Julia-1.5

Feedback ⓘ

released · Nov 4, 2026 MODEL ↗

Julia 1.5 is a release of the Julia programming language, a high-performance language for technical and scientific computing with a just-in-time compiler.

Julia-1.5 from Supersonic Labs was released on 8 Oct 2026. Open weights on Hugging Face.

Succeeds Julia-1.

WAS THIS USEFUL? 👍 👎

Supersonic Labs Julia-1.5-experts

Feedback ⓘ

released · Nov 4, 2026 MODEL ↗

Julia-1.5-experts from Supersonic Labs was released on 8 Oct 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

Cohere Models

Feedback ⓘ November 3, 2026 NOTES ↗

Cohere develops Command, Aya, Embed, Rerank, transcription, and document-understanding models.

Model Vault adds open model hosting with confidential computing, zero data retention by default, and an OpenAI-compatible Chat Completions API endpoint.

NEW INTEGRATION NEW SUPPORTED PLATFORM Model Vault Encrypted uses confidential computing so neither Cohere nor the cloud provider can access prompts or responses.

Open models are callable via the OpenAI-compatible Chat Completions API at the vault endpoint (`<YOUR_VAULT_ENDPOINT_URL>/v1`), keeping existing OpenAI client code intact.

Zero data retention is on by default — prompts and responses are not stored in Model Vault.

WAS THIS USEFUL? 👍 👎

DeepSeek V4 Flash 0731 Feedback ⓘ

price cut · Nov 3, 2026 MODEL ↗

DeepSeek V4 Flash 0731 is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4-flash-0731): $0.119 → $0.059 in, $0.238 → $0.30 out per million tokens (-50%), first seen 3 Nov 2026.

WAS THIS USEFUL? 👍 👎

Google Gemma 4 26B A4B Feedback ⓘ

price increase · Nov 3, 2026 MODEL ↗

Gemma 4 26B A4B is a multimodal model from Google.

Price increase on OpenRouter (google/gemma-4-26b-a4b-it): $0.0675 → $0.10 in, $0.225 → $0.30 out per million tokens (+48%), first seen 3 Nov 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 235B A22B Instruct 2507 Feedback

ⓘ price increase · Nov 3, 2026 MODEL ↗

Qwen3 235B A22B Instruct 2507 is a chat model from Alibaba.

Price increase on OpenRouter (qwen/qwen3-235b-a22b-2507): $0.09 → $0.14 in, $0.55 → $0.80 out per million tokens (+56%), first seen 3 Nov 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 Coder 30B A3B Instruct

Feedback ⓘ

price increase · Nov 3, 2026 MODEL ↗

Qwen3 Coder 30B A3B Instruct is a coding model from Alibaba.

Price increase on OpenRouter (qwen/qwen3-coder-30b-a3b-instruct): $0.07 → $0.15 in, $0.28 → $0.60 out per million tokens (+114%), first seen 3 Nov 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 Next 80B A3B Instruct

Feedback ⓘ

price increase · Nov 3, 2026 MODEL ↗

Qwen3 Next 80B A3B Instruct is a chat model from Alibaba.

Price increase on OpenRouter (qwen/qwen3-next-80b-a3b-instruct): $0.09 → $0.10 in, $1.10 → $1.10 out per million tokens (+11%), first seen 3 Nov 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · November 4, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue