THE AI TOOLCHAIN

NO. 044

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 2, 2026 · EVERY DAY

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

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 286 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 41 tools

▾ // THE BRIEF HIDE

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

Three unrelated things stand out. Ray now requires a cluster-wide auth token on every node by default, so an open dashboard port stops being a path into your GPUs. PydanticAI can review a tool call before it

vim ? help

executes. And Fireworks replaced managed RFT with full-parameter RL fine-tuning on the Training API, up to 524K context.

GOVERN

Stop shipping a Ray cluster that trusts anything on the network

Token authentication is enabled by default and required across all nodes as of 2.61 — Ray clusters historically assumed a trusted network, which is why an exposed head node has been a reliable way to get someone else's GPUs running your code. The hardening is now the default rather than a config you remember to set.

Ray

GOVERN

Block a dangerous agent tool call before it runs, not after

PydanticAI's ToolCallJudge inspects a proposed tool call pre-execution, moving the check off the human in the loop and ahead of the side effect. Copilot CLI's `copilot sandbox ca` suite manages proxy CA trust so sandboxed runs can reach an intercepting corporate proxy without blanket-trusting it, and read-only pipelines are reviewed automatically.

PydanticAI · GitHub Copilot CLI

BUILD

Fine-tune with RL on long contexts without a managed wrapper

Fireworks swapped Managed RFT for full-parameter Reinforcement Learning fine-tuning on the Training API with up to 524K context and on-policy distillation — relevant if your reward signal only shows up over a long trajectory that a short window truncates. Together Link wraps popular coding agents to route through Together-hosted 1M-context models with spend tracking attached.

Fireworks AI · Together AI

DEPLOY

Serve a prompt too long to fit one GPU's prefill

SGLang adds prefill context parallelism, splitting the prefill of a single long prompt across devices rather than queueing behind it, plus attention data parallelism via `--attn-dp-size` replacing the old `--dp-size --enable-dp-attention` pairing. Claude Code v2.1.287 makes 1M-token context the default for Opus 4.7+ and Fable across Bedrock, Vertex, Foundry and the Claude apps gateway.

SGLang

BUILD

◆ AI Coding Agents

yetone magpie

Feedback ⓘ v0.1.644 CODE ↗ LATEST STABLE

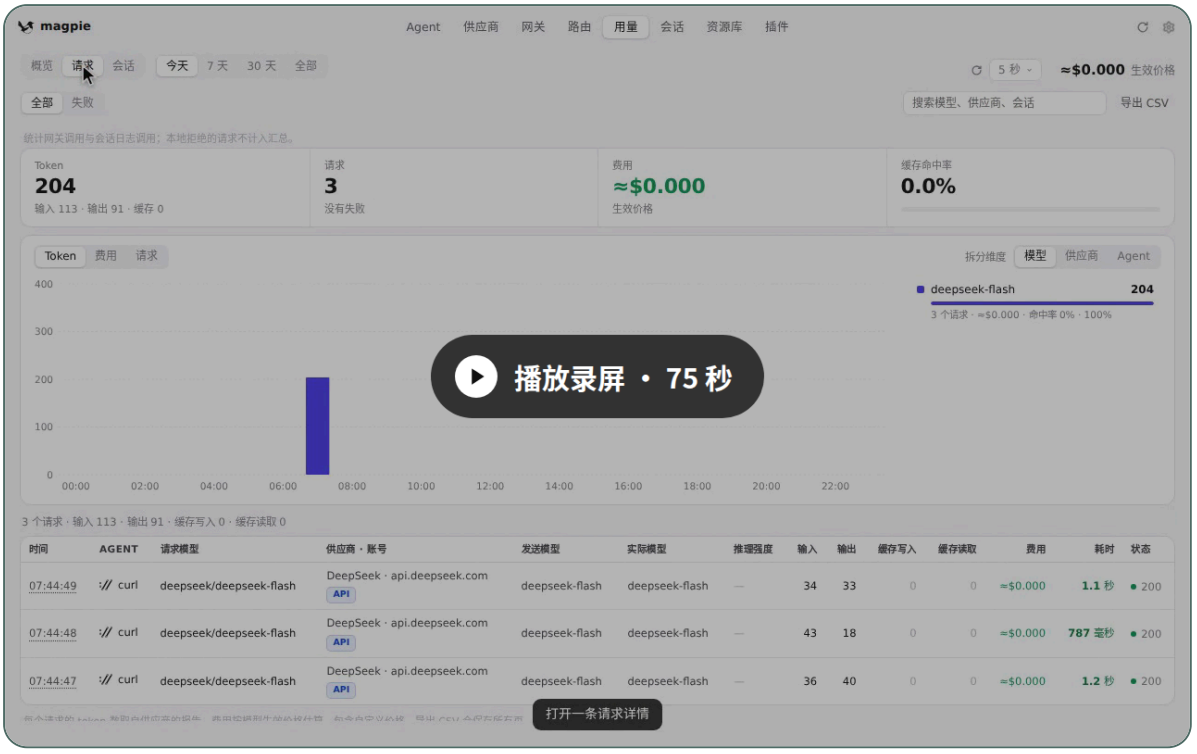
RANK

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.

Adds `pace` routing option to prioritize accounts by weekly quota remaining ÷ hours until reset, minimizing wasted allowances.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.1.644 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.644
```



└──▷ USE IT

Switch to weekly-pace routing so two Claude Max accounts each exhaust their weekly quota before reset, instead of one running dry while the other wastes its allowance.

env

```
routing=pace
```

Adds `routing=pace` (id `pace`) as a new routing option — `Weekly pace` — that sorts subscription accounts by `(100 − used%) ÷ hours until weekly reset`, putting the account with the most to lose at its reset window first.

Adds Allowance.Pace(model, now) method in `internal/provider/routing.go` returning (pace, due), selecting only budget windows (span ≥ 1 day or span = 0) for a given model and computing pace with a minimum denominator of 1 hour.

NEW INTEGRATION Kilo Gateway provider preset

UPDATED INTEGRATION ZCode and Pi project-level MCP server support

▾ **12 more releases in this issue** · 2026-10-02

v0.1.641 CODE ↗ STABLE

magpie v0.1.641: dsh route auto-repair every 30 s, bulk session deletion, and smarter In-order account rotation

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.641 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.641
```

Adds periodic re-write of dsh's magpie route every 30 s while magpie serves the gateway via `dshRouteAgain`, so stale routes left by dsh's own Models page or an older magpie no longer break all sessions in a profile until restart.

Enables deleting every session under a project folder at once from the folder's row on the Sessions page.

Keeps Codex and Claude Code signed in to the first account under the `In order` routing mode until that account is fully used up or refused, rather than switching at 98% usage.

Allows an incomplete sign-in flow for a second account to be dismissed from beside `Open again` or with the editor, preventing the browser-login window from getting stuck open.

v0.1.638 CODE ↗ STABLE

Magpie gains Factory account import via API key, provider-order sync, and durable Claude usage snapshots with `asOf` timestamps.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.638 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.638
```

└──▷ **USE IT**

Import a Factory account by pasting its API key from a `.env` file — magpie finds and validates the `fk-…` key automatically.

.env

```
FACTORY_API_KEY=fk-yourKeyHere
```

Check Claude quota in CI when `/usage` may be temporarily unavailable — the `asOf` field in JSON output tells you how stale the cached reading is.

```
$ magpie accounts claude --json
```

Adds Factory account import by API key (`fk-…`), parsed from pasted text, a `.env` line (`FACTORY_API_KEY=fk-…`), or JSON, with each key validated via `whoami` before being added.

Adds `asOf` field to the public quota and accounts JSON output when a dated Claude usage snapshot is displayed in place of a live reading.

Preserves the last Claude usage snapshot — with its original timestamp, percentages, and reset times — when `/usage` returns only the subscription notice or no text, surfacing the dated reading in `magpie quota` and `magpie accounts` .

Syncs provider order from the Providers tab so that a machine receiving a WebDAV sync tries providers in the same sequence as the sending machine.

v0.1.637 CODE ↗ STABLE

Magpie v0.1.637 adds Pencil (pen.dev) agent integration, writing magpie models into `~/.pencil/models.json` .

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.637 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.637
```

Adds Pencil (pen.dev) as a supported agent, with magpie automatically managing its custom models in `~/.pencil/models.json` under the `providers` key, using the `openai-completions` API at magpie's gateway `/v1`.

Adds gateway logging when a Claude Code run is released because earlier messages changed while waiting for tool results — one log line per occurrence helps diagnose clients that rewrite conversation history and lose cached prompts each tool round.

Portable magpie on Windows now keeps WebView2's profile in `data\webview2` and omits the Start-menu shortcut; installed builds create the shortcut once and never recreate it after the user deletes it.

v0.1.636 CODE ↗ STABLE

magpie v0.1.636 extends WSL CLI setup to OpenCode, Kimi Code, omp and more, and fixes ClinePass model probing.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.636 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.636
```

Extends WSL distro setup to OpenCode (`.config/opencode/opencode.json`), Kimi Code (`.kimi-code/config.toml`), omp (`.omp/agent/models.yml`), crush (`.config/crush/crush.json`), hermes (`.hermes/config.yaml`), grok (`.grok/config.toml`), droid (`.factory/settings.json`), fx (`.fx/settings.json`), commandcode (`.commandcode/providers.json`), minimax-code (`.minimax/config.yaml`), muse (`.config/muse/settings.json`), qoder (`.qoder/settings.json`), and qoder-cn (`.qoder-cn/settings.json`) — previously only Codex, Pi, OmO, and Claude Code were configured in WSL distros on Windows.

Tests ClinePass free models the same way Cline's own clients do (streamed, with room to respond), fixing silent 500 failures when probing ClinePass free models via right-click.

v0.1.635 CODE ↗ STABLE

Adds Kilo Gateway free-model preset, ChatGPT effort-level account fallback with 1-hour memory, Claude Code logout scoping, and content-hash cache-busting for the web UI.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.635 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.635
```

Adds a Kilo Gateway provider preset that accesses Kilo Code's free models with no API key required.

Adds reasoning-effort-aware account fallback for ChatGPT: when an account's plan rejects a requested effort level, the gateway tries another account and remembers the refusal for one hour before retrying.

Scopes Claude Code `/logout` to only the signed-in account, preventing that account from being served as a substitute after logout.

Adds content-hash-based cache-busting for magpie web's scripts and styles, preventing a cache from serving a stale `app.js` under a newer page.

v0.1.634 CODE ↗ STABLE

Adds a Free only bulk-select button to the provider model list, appearing only when the list has both free and paid models.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.1.634 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.634
```

v0.1.633 CODE ↗ STABLE

magpie v0.1.633 adds per-provider max concurrency queuing and DeepSeek Responses API web_search support

└──▷ GET THIS VERSION

```
$ git clone --branch v0.1.633 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.633
```

Adds `MaxConcurrency` setting per provider so requests beyond the limit queue in arrival order instead of being dropped or rerouted — excess requests are held in `Try.Queued` and dispatched as slots free.

Supports the `web_search` hosted tool on DeepSeek's `api.deepseek.com/v1/responses` Responses API endpoint, which magpie previously dropped.

Adds `Add provider` sheet as a floating overlay so opening and closing it no longer shifts the provider list's scroll position.

v0.1.630 CODE ↗ STABLE

magpie v0.1.630 adds portable mode and ClinePass free-model support with correct desktop headers

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.630 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.630
```

└──▷ **TRY IT**

Run magpie in portable mode on a USB drive or shared workstation so no config leaks into the user profile

```
$ mkdir /path/to/magpie-dir/data
# magpie will now read/write all its own files under
/path/to/magpie-dir/data/
/path/to/magpie-dir/magpie
```

Adds portable mode: placing a folder named `data` or a file named `.portable` beside the magpie executable redirects all magpie-owned files into that folder (and caches into `data/cache`), keeping nothing in the user profile — mirroring VS Code's portable mode.

Adds `ClinePass` provider preset that enumerates Cline's real free models and authenticates as Cline's desktop app (version `3.5.54`) using the correct `X-CLIENT-TYPE: cline-desktop` and related headers, fixing 403 errors on `cline-free/` model IDs.

Rate-limit handling now retries the last remaining provider on a 429 even when no `Retry-After` header is present, instead of giving up.

v0.1.629 CODE ↗ STABLE

Magpie v0.1.629 lets library MCP servers be scoped per-project, written into each agent's project-level config file.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.1.629 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.629
```

Writes library MCP servers into per-project config files for each supported agent: `.mcp.json` (Claude), `.codex/config.toml` (Codex), `.gemini/settings.json` (Gemini CLI), `.cursor/mcp.json` (Cursor), and `opencode.json` (OpenCode), alongside any servers already in those files.

Adds project-level MCP server management to the library UI, showing projects in the MCP tab the same way they already appeared in the Skills tab.

v0.1.618 CODE ↗ STABLE

Adds PipeLLM as a built-in provider preset, routing OpenAI, Anthropic, and Gemini models via a single API key.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.1.618 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.618
```

Adds `type_target` field support in live model catalog entries, mapping `openai` models to `chat` / `responses` APIs, `anthropic` models to the `anthropic` API, and `gemini` models to `chat`.

v0.1.617 CODE ↗ STABLE

magpie web is now usable on phone screens, with improved layout for Agent fields, Usage details, and Library controls.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.1.617 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.617
```

Makes magpie web usable on phone and touch-tablet screens, with full-row model display per Agent, wrapped effort/extras, bounded pickers, and readable Usage totals and chart dates.

Replaces the inline `Add provider` drawer with a floating overlay panel that preserves the existing provider list position and scroll height; supports close button, overlay tap, and Esc to dismiss.

v0.1.616 CODE ↗ STABLE

Magpie v0.1.616 adds transparent proxying for OpenCode Zen free-tier models and automatic Mistral thinking-field stripping.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.616 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.616
```

Supports routing requests to OpenCode Zen free-tier models (e.g. `mimo-v2.6-flash-free`) from Claude Code, Codex, and other clients by translating tool names and headers to satisfy Zen's OpenCode-only restrictions.

Automatically strips earlier assistant-turn `thinking` fields before forwarding to Mistral, then retries without them when Mistral rejects the field with a 422 `extra_forbidden` error.

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.162.0-alpha.5 NOTES ↗

CODE ↗ RANK

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

> OpenAI Codex CLI's rust v0.160–v0.162 alpha releases headline per-turn Cyber access program selection via a new `--cyber-access-program` flag and TypeScript SDK type, alongside MCP-based worktree management tools and a batch of TUI refinements — configurable mouse scroll speed, Guardian review context, history pagination, X11 middle-click paste, and projectless sessions.

└──▷ **WHAT SHIPPED · 14 FEATURES** most completely described first ? what's the number?

01 Configurable transcript mouse scroll speed

BREAKING

90

Adds `tui.mouse_scroll_speed` config key as a finite positive multiplier for transcript mouse wheel scrolling, covering overlays and session previews. The default changes from 3 rows to 1 row per wheel event; set it to `3.0` to restore the previous speed, and fractional values such as `0.5` accumulate across events.

Restore the previous 3-row-per-event scroll speed, or tune it to a slower fractional rate for fine-grained scrolling through long transcripts.

yaml

```yaml
tui:
  mouse_scroll_speed: 3.0
```

— Exact config key, default change, and restore value given. rust-v0.162.0-alpha.5

02 Cyber access program selection via flag and SDK

NEW

90

Adds a `--cyber-access-program` flag to `codex exec` and a `cyberAccessProgram` turn option with a `CyberAccessProgram` type in the TypeScript SDK for per-turn Cyber access program selection, plus support for explicit cyber access programs authenticated with OpenAI API keys.

Run a single exec turn against the red-team Daybreak program without changing your saved program preference.

```
$ codex exec --cyber-access-program daybreak_red 'Review this codebase for privilege escalation paths'
```

— Named flag, SDK type, and a runnable example command. `rust-v0.162.0-alpha.1` `rust-v0.162.0-alpha.2`

03 **Worktree management tools via MCP** NEW 75

Exposes `create_worktree`, `get_worktree_creation_status`, and `list_worktrees` as MCP tools in the TUI when the worktrees feature is enabled for a trusted local project with attachment storage available.

— Named MCP tool surfaces with enablement conditions. `rust-v0.162.0-alpha.3`

04 **MCP server login from TUI** NEW 75

Adds a `/mcp login <name>` command to authenticate an MCP server by name directly from the TUI without leaving the session.

Authenticate an MCP server by name directly from the TUI without leaving your session.

```
$ /mcp login <name>
```

— Runnable command shown, but little further mechanism. `rust-v0.162.0-alpha.2`

05 **X11 middle-click paste for transcript text** NEW 65

Adds X11 primary selection and middle-click paste support for transcript text in fullscreen mode on supported local Linux X11 terminals.

— Precisely scoped to platform and mode, no command needed. `rust-v0.160.0`

06 **TUI mode reporting in codex doctor** NEW 60

`codex doctor` now reports the configured TUI mode (`fullscreen` or `scrollback`), reflecting the `tui.fullscreen_transcript` setting.

— Names command and setting but no deeper mechanism. `rust-v0.162.0-alpha.5`

THINNER COVERAGE BELOW

07 **Projectless sessions with workspace defaults** NEW 55

Supports starting sessions outside a project using workspace defaults when policy permits, and restores saved permissions when resuming such sessions.

— Explains behaviour but no named flag or config key. `rust-v0.160.0`

08 **Server-synced TUI permission shortcuts** IMPROVED 50

Syncs TUI permission shortcuts to the connected server's catalog so model-specific auto-review requirements and server restrictions are respected consistently alongside the permission picker.

– Explains behaviour but no named flag or config key. `rust-v0.162.0-alpha.3`

09 **History pagination in agent command center** IMPROVED

50

Adds history pagination to the agent command center with a keyboard-accessible Show more action.

– Names the UI action but not an exact navigation path. `rust-v0.160.0`

10 **Guardian review context retrieval** NEW

45

Adds opt-in Guardian review capabilities to retrieve earlier user instructions and include context from agent handoffs.

– Describes capability but no config key or command given. `rust-v0.160.0`

11 **Copy selected file paths as plain text** IMPROVED

30

The TUI now copies selected file paths as plain text.

– Bare description with no shortcut or mechanism named. `rust-v0.162.0-alpha.2`

12 **MCP tool pagination in legacy protocol mode** IMPROVED

30

Supports MCP tool pagination when operating in legacy protocol mode.

– Names the mode but gives no usage detail. `rust-v0.162.0-alpha.1`

13 **Renewable EMA HTTP authentication and credential versioning** NEW

30

Adds renewable EMA HTTP authentication along with credential versioning.

– Names the mechanism but offers no further explanation. `rust-v0.162.0-alpha.1`

14 **Experimental bundled tools flag for login shells** NEW

25

Adds an experimental flag that enables bundled tools when Codex CLI runs in login shells.

– Flag name and mechanism not specified. `rust-v0.162.0-alpha.2`

└──▷ **BREAKING ON UPGRADE**

! The default transcript mouse scroll speed changes from 3 rows to 1 row per wheel event; set `tui.mouse_scroll_speed` to `3.0` in config to restore the previous behavior.

rust-v0.162.0-alpha.5 NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI gains configurable transcript mouse scroll speed and TUI mode reporting in `codex doctor`.

└──▷ **GET THIS VERSION**

```
$ git clone --branch rust-v0.162.0-alpha.5
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.162.0-alpha.5
```

└──▷ **USE IT**

Restore the previous 3-row-per-event scroll speed, or tune it to a slower fractional rate for fine-grained scrolling through long transcripts.

```yaml
tui:
  mouse_scroll_speed: 3.0
```

Adds `tui.mouse_scroll_speed` config key as a finite positive multiplier for transcript mouse wheel scrolling (overlays and session previews included); default changes from 3 rows to 1 row per wheel event — set to `3.0` to restore previous speed; fractional values (e.g. `0.5`) accumulate across events.

Reports configured TUI mode (`fullscreen` or `scrollback`) in `codex doctor` output, reflecting the `tui.fullscreen_transcript` setting.

└──▷ **BREAKING ON UPGRADE**

! The default transcript mouse scroll speed changes from 3 rows to 1 row per wheel event; set `tui.mouse_scroll_speed` to `3.0` in config to restore the previous behavior.

rust-v0.162.0-alpha.3 NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI gains managed worktree tools via MCP and unified server permission catalog for TUI shortcuts.

└──▷ **GET THIS VERSION**

```
$ git clone --branch rust-v0.162.0-alpha.3
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.162.0-alpha.3
```

Exposes `create_worktree`, `get_worktree_creation_status`, and `list_worktrees` as MCP tools in the TUI when the worktrees feature is enabled for a trusted local project with attachment storage available.

Syncs TUI permission shortcuts to the connected server's catalog so model-specific auto-review requirements and server restrictions are respected consistently alongside the permission picker.

rust-v0.162.0-alpha.2 NOTES ↗ CODE ↗ PRE-RELEASE

OpenAI Codex CLI: Enables multi-agent V2 and Ultra reasoning on Amazon Bedrock.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.162.0-alpha.2
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.162.0-alpha.2
```

└──▷ TRY IT

Authenticate an MCP server by name directly from the TUI without leaving your session.

```
$ /mcp login <name>
```

Adds GPT-6.1 Sol as the default model in the standard and Bedrock catalogs.

Adds an experimental flag for bundled tools in login shells.

Supports explicit cyber access programs with OpenAI API keys.

Copies selected file paths as plain text in the TUI.

rust-v0.162.0-alpha.1 NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI adds `--cyber-access-program` flag for per-turn selection of Cyber access programs in `codex exec` and the TypeScript SDK.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.162.0-alpha.1
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.162.0-alpha.1
```

└──▷ **TRY IT**

Run a single exec turn against the red-team Daybreak program without changing your saved program preference.

```
$ codex exec --cyber-access-program daybreak_red 'Review this
codebase for privilege escalation paths'
```

Exposes `cyberAccessProgram` turn option and `CyberAccessProgram` type in the TypeScript SDK for per-turn Cyber access program selection.

Supports MCP tool pagination in legacy protocol mode.

Adds renewable EMA HTTP authentication and credential versioning.

`rust-v0.160.0` NOTES ↗ CODE ↗ **LATEST** STABLE

Codex CLI v0.160.0 adds agent command center history pagination, X11 middle-click paste, projectless sessions, and opt-in Guardian review context.

└──▷ **GET THIS VERSION**

```
$ git clone --branch rust-v0.160.0
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.160.0
```

Adds opt-in Guardian review capabilities to retrieve earlier user instructions and include context from agent handoffs.

Adds history pagination to the agent command center with a keyboard-accessible Show more action.

Adds X11 primary selection and middle-click paste support for transcript text in fullscreen mode on supported local Linux X11 terminals.

Supports starting sessions outside a project with workspace defaults when policy permits, and restores saved permissions when resuming.

WAS THIS USEFUL?  

Cline

Feedback ⓘ desktop-v0.0.42 NOTES ↗ CODE ↗ RANK

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

> Cline's biggest work this window was an agent-team SQLite storage overhaul with a new vacuum() method and capped event tables, alongside a redesigned Connectors tab, multi-tier cloud session models, and a new `/compact` slash command.

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

01 Agent team SQLite storage overhaul IMPROVED 95

Agent teams no longer re-save the full team state on every streamed chunk or heartbeat, preventing slowdowns and unbounded growth of `~/.cline/data/db/teams.db`; existing team data is compacted automatically on first open. Storage was migrated to schema v2, capping `team_events` at 2000 rows and 30 days per team and writing only changed entities in batched transactions every ~300 ms, and SqliteTeamStore.vacuum() can be called explicitly to compact storage and reclaim freed space (one observed `teams.db` was measured at 1.66 GB) after the one-time migration.

After upgrading, call vacuum() to trigger the one-time schema v2 migration and reclaim disk space from bloated team storage (one `teams.db` was measured at 1.66 GB).

```typescript
const freed = await store.vacuum();
console.log(`Freed ${freed} bytes`);
```

— Names exact file path, method, caps, and runnable example sdk/sdk/v0.0.90 cli-v3.0.68

02 Connectors tab redesign in Customize IMPROVED 70

The `Customize > Connectors` tab gained a Browse section listing the full connector catalog in a two-column grid with its own search box, became the first and default tab in Customize, and now shows connector brand logos; a sign-in card or rollout notice with a Check again button appears when connectors are unavailable to signed-out or non-beta users. It was then redesigned to match the Skills, Plugins, and MCP tabs, adding a refresh button, an Installed section, and a new Suggested section with ready-made connector combos (e.g. incident root-cause analysis, day organization), each with an example prompt and one-click install.

— Several UI additions named but no config or API surface desktop-v0.0.41 desktop-v0.0.42

03 **/compact slash command in desktop composer** NEW 70

Adds a `/compact` slash command to the desktop app composer that summarizes the current conversation to free up context, using the session's provider settings.

When a long session is consuming context headroom, run `/compact` in the composer to collapse the transcript before continuing.

```
$ /compact
```

— Names exact command with a runnable usage example desktop-v0.0.42

04 **Multi-tier model picker for cloud sessions** NEW 65

Cloud sessions now support Cline Usage-Billing, `ClinePass`, and `ClineFree` model tiers in the model picker, with session start held until a model is ready and a retry option if the model list fails to load.

— Names tiers and behavior but no exact command or config desktop-v0.0.41

THINNER COVERAGE BELOW

05 **Default model updates across providers** IMPROVED 55

Default models updated for DigitalOcean (GPT-6.1 Sol), GMI Cloud (Qwen 3.8 Flash), NanoGPT (Pareto 26.10 Preview), Nvidia (DeepSeek V4.1 Flash), and Ofox (GPT-6.1 Sol) in the refreshed model catalog.

— Lists exact providers and models but no usage steps desktop-v0.0.41

06 **Marketplace item type display and filters** IMPROVED 50

Marketplace rows now display item type to distinguish same-named entries (e.g. the Figma MCP server vs. the Figma connector), and category filters are consolidated into a single dropdown.

— Describes UI change without exact navigation steps desktop-v0.0.41

07 **Active app sessions highlighted in sidebar** IMPROVED 30

Active app sessions are now highlighted in the sidebar while open, matching the existing behavior for sessions reopened from history.

— Minor UI tweak with no named surface desktop-v0.0.41

> desktop-v0.0.42 NOTES ↗ CODE ↗ LATEST STABLE
>
> Cline desktop v0.0.42 adds a `/compact` slash command and a browsable connector catalog inside Customize.

└──▷ GET THIS VERSION

```
$ git clone --branch desktop-v0.0.42
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.42
```

└──▷ TRY IT

When a long session is consuming context headroom, run `/compact` in the composer to collapse the transcript before continuing.

```
$ /compact
```

Adds `/compact` slash command to the desktop app composer, which summarizes the current conversation to free up context using the session's provider settings.

Adds a `Browse` section to `Customize > Connectors` listing the full connector catalog in a two-column grid with its own search box, so connectors can be discovered without visiting the Marketplace.

Makes `Connectors` the first and default tab in `Customize`, with separate search boxes for Installed and Browse sections.

Shows connector brand logos in both the Customize view and Marketplace rows.

Displays an explanatory sign-in card or rollout notice with a `Check again` button on the Connectors tab when connectors are unavailable to the signed-out or non-beta user.

desktop-v0.0.41 NOTES ↗ CODE ↗ STABLE

Cline desktop v0.0.41 adds multi-model cloud sessions, a redesigned Connectors tab with Suggested combos, and draft-prompt persistence across sessions.

└──▷ GET THIS VERSION

```
$ git clone --branch desktop-v0.0.41
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.41
```

Cloud sessions now support `Cline Usage-Billing`, `ClinePass`, and `ClineFree` model tiers in the model picker, with session start held until a model is ready and a retry option if the model list fails to load.

Customize → Connectors tab is redesigned to match Skills, Plugins, and MCP tabs, adding search, a refresh button, an Installed section, and a new Suggested section with ready-made connector combos (e.g. incident root-cause analysis, day organization), each with an example prompt and one-click installs.

Marketplace rows now display item type to distinguish same-named entries (e.g. the Figma MCP server vs. the Figma connector), and category filters are consolidated into a single dropdown.

Active app sessions are now highlighted in the sidebar while open, matching the existing behavior for sessions reopened from history.

Default models updated for DigitalOcean (GPT-6.1 Sol), GMI Cloud (Qwen 3.8 Flash), NanoGPT (Pareto 26.10 Preview), Nvidia (DeepSeek V4.1 Flash), and Ofox (GPT-6.1 Sol) in the refreshed model catalog.

cli-v3.0.68 NOTES ↗ CODE ↗ STABLE

Agent teams stay fast in long sessions; default models updated for DigitalOcean, GMI Cloud, NanoGPT, Nvidia, and Ofox providers.

└──▷ GET THIS VERSION

```
$ git clone --branch cli-v3.0.68 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout cli-v3.0.68
```

Agent teams no longer re-save the full team state on every streamed chunk, preventing slowdowns and unbounded growth of `~/.cline/data/db/teams.db`; existing team data is compacted automatically on first open.

sdk/sdk/v0.0.90 NOTES ↗ CODE ↗ STABLE

Cline SDK v0.0.90 overhauls agent-team SQLite storage with schema v2, caps event tables, and exposes SqliteTeamStore.vacuum() for space reclamation.

└──▷ GET THIS VERSION

```
$ git clone --branch sdk/sdk/v0.0.90
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout sdk/sdk/v0.0.90
```

└──▷ USE IT

After upgrading, call vacuum() to trigger the one-time schema v2 migration and reclaim disk space from bloated team storage (one `teams.db` was measured at 1.66 GB).

```typescript
const freed = await store.vacuum();
console.log(`Freed ${freed} bytes`);
```

Introduces SqliteTeamStore.vacuum() — call it explicitly to compact team storage and get back the freed space after the one-time schema v2 migration.

Migrates SQLite team storage to schema v2, capping `team_events` at 2000 rows and 30 days per team, and writing only changed entities in batched transactions every ~300 ms instead of re-saving full state on every streamed chunk or heartbeat.

WAS THIS USEFUL? 👍 👎

Earendil Works Pi

Feedback ⓘ v1.0.0 NOTES ↗ RANK

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

> Pi's v1.0.0 release is the centerpiece of this window, adding image generation via OpenRouter models, a fullscreen-by-default TUI, hardened MCP OAuth with per-server credentials, and a Radius login option, alongside a ~40% reduction in codemode prompt tokens and several codemode breaking changes.

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

01 **Fullscreen TUI by default with regular-mode opt-out** BREAKING

95

The default TUI mode is now fullscreen; existing setups that relied on normal terminal scrollback must set `tuiMode` to `"regular"` in config or pass `--tui-mode regular`. Adds `quietStartup: "header"` to keep the version/key-hint header while hiding the model scope line and loaded-resource listing.

Keep normal terminal scrollback on a system where fullscreen TUI breaks your workflow.

```
$ pi --tui-mode regular
```

Suppress verbose startup output in CI while retaining the version and key-hint header.

```yaml
quietStartup: "header"
```

Keep the old scrollback-based terminal layout when upgrading to v1.0.0, where fullscreen is now the default.

```
$ pi --tui-mode regular
```

Suppress the verbose startup resource listing on a shared/monitored terminal while keeping the version banner and key hints.

```yaml
quietStartup: "header"
```

— Exact flag, config key and migration path given with runnable examples v1.0.0

02 **Image generation via OpenRouter models** NEW 90

Adds models.getAvailableOfType("image") to list available image models (e.g. `google/gemini-2.5-flash-image`, `black-forest-labs/flux.2-pro`) from the OpenRouter provider, and models.generateImages(model, { input }) to generate images from a prompt or `{ type: "image", data, mimeType }` input blocks, returning base64 image blocks in `result.output` — supporting image editing and reference-based generation. Extensions can call ctx.modelRegistry.generateImages(), codemode scripts can call models.generateImages() with session credentials, and image model usage is tracked and added to the tool result like classifier calls.

— Names exact functions, models and data shapes; no runnable CLI example given v1.0.0

03 **MCP OAuth hardening and per-server credentials** BREAKING 90

Improves MCP OAuth with RFC 9207 `iss` parameter validation, per-server credential storage keyed by server name and URL, and step-up sign-in that preserves previously granted scopes. Adds `--oauth-client-name` to `pi mcp add` so Pi registers under a known OAuth client, and an `authServerMetadataUrl` key under `oauth` in `mcp.json` to override a server's advertised authorization-server metadata. Breaking change: credentials previously stored by URL alone are moved to the first server using that URL; servers sharing a URL that signed in with different accounts will need to re-authenticate.

Add an MCP server that only accepts a known OAuth client name, so Pi registers under the right identity during OAuth.

```
$ pi mcp add https://my-mcp-server.example.com --oauth-client-name my-approved-client
```

Override the OAuth authorization server metadata for an MCP server that advertises an incorrect or missing authorization server URL.

```yaml
oauth:
  authServerMetadataUrl: "https://auth.example.com/.well-known/oauth-authorization-server"
```

— Names flag, config key and file, plus concrete migration consequence v1.0.0

04 **Radius login option for MCP** NEW 70

Adds Radius as a `/login` option, signing in and optionally configuring the Radius MCP server in the global `mcp.json` with `"auth": { "provider": "radius" }` in one

step.

— Names the UI command and exact config fragment it writes v1.0.0

05 **Codemode exposure alias and tool-probing change** BREAKING

70

`codemode-deferred` exposure is now an alias for the default exposure — scripts that depended on `codemode-deferred` as a distinct exposure value should migrate to the plain default exposure. Codemode scripts that probed for a tool with `typeof tools.name` must now use `"name" in tools`.

— Names both exact old and new syntax so a migration is clear v1.0.0

06 **/reload picks up newly added tools** IMPROVED

65

Makes `/reload` enable tools newly added to the `tools` setting, while respecting `--no-tools` and `--no-builtin-tools` overrides.

— Names the exact command and flags whose behavior it respects product docs

07 **Codemode prompt token reduction and clearer errors** IMPROVED

65

Reduces codemode prompt tokens by ~40% (GPT-5.6 requests shrink from ~5,300 to ~3,300 tokens), with errors that name close matches and suggest recovery actions.

— Concrete before/after numbers but no action for the reader to take product docs

THINNER COVERAGE BELOW

08 **Copy-code login for Anthropic in headless setups** NEW

50

Adds a copy-code login method to Anthropic `/login` for headless setups where the browser runs on a different machine.

— Describes the method but not the exact steps or flag v1.0.0

└──▷ **BREAKING ON UPGRADE**

! The default TUI mode is now fullscreen; existing setups that relied on normal terminal scrollback must set `tuiMode` to `"regular"` or pass `--tui-mode regular`.

! MCP OAuth credentials previously stored by URL alone are moved to the first server that uses that URL; servers sharing the same URL that previously shared credentials now have per-server credentials keyed by server name and URL.

! `codemode-deferred` exposure is now an alias for the default exposure — scripts that depended on `codemode-deferred` as a distinct exposure value should migrate to the plain default exposure.

! The default TUI mode is now fullscreen; set `tuiMode` to `"regular"` or pass `--tui-mode regular` to restore the terminal's normal scrollback.

! MCP OAuth credentials previously stored by URL alone are moved to the first server that uses that URL; servers sharing a URL that signed in with different accounts will need to re-authenticate.

! Codemode scripts that probed for a tool with `typeof tools.name` must now use `"name" in tools`.

v1.0.0 NOTES ↗ LATEST STABLE

Pi v1.0.0 adds image generation in codemode, Radius login, MCP OAuth hardening, and new `tuiMode` / `quietStartup` config keys.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.0.0 https://github.com/earendil-works/pi.git
# already have the repo? check out this version:
$ git checkout v1.0.0
```

└──▷ **TRY IT**

Keep the old scrollback-based terminal layout when upgrading to v1.0.0, where fullscreen is now the default.

```
$ pi --tui-mode regular
```

Suppress the verbose startup resource listing on a shared/monitored terminal while keeping the version banner and key hints.

```yaml
quietStartup: "header"
```

Override the OAuth authorization server metadata for an MCP server that advertises an incorrect or missing authorization server URL.

```mcp.json
oauth:
  authServerMetadataUrl: "https://auth.example.com/.well-known/oauth-authorization-server"
```

Adds `tuiMode` config key (set to `"regular"`) and `--tui-mode regular` CLI flag to opt out of the new fullscreen-by-default TUI.

Adds `quietStartup: "header"` config setting to keep the version/key-hint header while hiding the model scope line and loaded-resource listing.

Adds models.generateImages() to codemode scripts, calling image models (e.g. OpenRouter) with session credentials and returning base64 image blocks; extensions can call ctx.modelRegistry.generateImages().

Adds Radius as a `/login` option, signing in and optionally configuring the Radius MCP server in the global `mcp.json` with `"auth": { "provider": "radius" }` in one step.

Adds a copy-code login method to Anthropic `/login` for headless setups where the browser runs on a different machine.

+1 more

└─▷ **BREAKING ON UPGRADE**

- ! The default TUI mode is now fullscreen; set `tuiMode` to `"regular"` or pass `--tui-mode regular` to restore the terminal's normal scrollback.
- ! MCP OAuth credentials previously stored by URL alone are moved to the first server that uses that URL; servers sharing a URL that signed in with different accounts will need to re-authenticate.
- ! Codemode scripts that probed for a tool with `typeof tools.name` must now use `"name" in tools`.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.287 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.287 defaults Opus 4.7+ and Fable to a 1M-token context window across Bedrock, Vertex, Foundry and the Claude apps gateway, introduces a Claude Mods plugin framework with a built-in You should know safety agent, and adds MCP URL-prompt support, an agents-view session filter, an OpenTelemetry prompt_text field, and a self-hosted-runner gh api command.

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

01 1M context window default for Opus 4.7+ and Fable

BREAKING

90

Changes Opus 4.7+ and Fable to use a 1M context window by default on Bedrock, Vertex, Foundry, and the Claude apps gateway (removing the `[1m]` suffix); set `CLAUDE_CODE_DISABLE_1M_CONTEXT=1` to keep the 200K window.

Preserve the 200K context window on Bedrock/Vertex instead of the new 1M default, e.g. to stay within cost or quota limits.

```
$ CLAUDE_CODE_DISABLE_1M_CONTEXT=1 claude
```

— Exact env var and runnable command to revert default v2.1.287

02 URL prompts from MCP servers

BREAKING

84

Adds support for URL prompts from MCP servers on the 2025-11-25 protocol (e.g. sign-in flows); servers that no longer connect after this update require `"bareElicitationCapability": true` in their MCP config entry.

Allow an MCP server that presents URL prompts (2025-11-25 protocol) to connect after this update by adding the required capability flag to its config entry.

```json
{
  "mcpServers": {
    "my-server": {
      "command": "npx",
      "args": ["my-mcp-server"],
      "bareElicitationCapability": true
    }
  }
}
```

— Full config snippet and exact required flag given v2.1.287

03 Claude Mods framework and 'You should know' agent NEW

73

Adds Claude Mods, letting plugins modify deeper Claude behavior. Ships You should know as a built-in mod where a side agent watches and flags things you or Claude might miss; enable with `/plugin enable cc-plugin-you-should-know@builtin` (requires first-party sessions with telemetry on).

Enable the You should know background safety agent to catch things you or Claude might miss during a session.

```
$ /plugin enable cc-plugin-you-should-know@builtin
```

— Exact enable command given but Mods mechanism stays vague v2.1.287

04 Session filter in agents view NEW

70

Adds `n:<text>` filter to the agents view that matches session names and tasks, shows matches in collapsed sections, and opens the first match with Enter.

— Exact filter syntax and keybinding named v2.1.287

THINNER COVERAGE BELOW

05 Built-in gh api for self-hosted runners NEW

59

Self-hosted runner adds a built-in `gh api` (REST only) for sessions using Anthropic-managed git on macOS and Linux machines where the GitHub CLI is not installed.

— Names surface but gives no example invocation v2.1.287

06 prompt_text field in OpenTelemetry user_prompt event NEW

57

Adds `prompt_text` to the OpenTelemetry `user_prompt` event, a copy of `prompt` for backends that nest dotted keys; drop or mask it wherever you drop or mask `prompt`.

— Names exact field and event but no config example v2.1.287

07 **Windows shell-tool denial warning** IMPROVED 43

Windows: adds a startup warning when denying the Bash tool also disables the PowerShell tool, leaving Claude with no shell tool.

— Describes behavior but no remediation steps given v2.1.287

└──▷ BREAKING ON UPGRADE

! MCP servers using the 2025-11-25 protocol that present URL prompts (e.g. for sign-in) may fail to connect after this update unless `"bareElicitationCapability": true` is added to their MCP config entry.

! Opus 4.7+ and Fable now default to a 1M context window on Bedrock, Vertex, Foundry, and the Claude apps gateway (the `[1m]` suffix is removed); set `CLAUDE_CODE_DISABLE_1M_CONTEXT=1` to restore the 200K window.

v2.1.287 NOTES ↗ CODE ↗ LATEST STABLE

Claude Code v2.1.287 adds Claude Mods, a You should know safety agent, OpenTelemetry prompt_text field, and MCP URL prompts.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.287 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.287
```

└──▷ TRY IT

Enable the You should know background safety agent to catch things you or Claude might miss during a session.

```
$ /plugin enable cc-plugin-you-should-know@builtin
```

Preserve the 200K context window on Bedrock/Vertex instead of the new 1M default, e.g. to stay within cost or quota limits.

```
$ CLAUDE_CODE_DISABLE_1M_CONTEXT=1 claude
```

Allow an MCP server that presents URL prompts (2025-11-25 protocol) to connect after this update by adding the required capability flag to its config entry.

```json
{
  "mcpServers": {
    "my-server": {
      "command": "npx",
      "args": ["my-mcp-server"],
      "bareElicitationCapability": true
    }
  }
}
```

Adds `prompt_text` to the OpenTelemetry `user_prompt` event, a copy of `prompt` for backends that nest dotted keys; drop or mask it wherever you drop or mask `prompt`.

Adds support for URL prompts from MCP servers on the 2025-11-25 protocol (e.g. sign-in flows); servers that no longer connect after this update require `"bareElicitationCapability": true` in their MCP config entry.

Adds `n:<text>` filter to the agents view that matches session names and tasks, shows matches in collapsed sections, and opens the first match with Enter.

Adds Claude Mods: plugins may now modify deeper Claude behavior.

Adds You should know, a built-in mod where a side agent watches and flags things you or Claude might miss; enable with `/plugin enable cc-plugin-you-should-know@builtin` (first-party sessions with telemetry on).

+3 more

└─▷ BREAKING ON UPGRADE

! MCP servers using the 2025-11-25 protocol that present URL prompts (e.g. for sign-in) may fail to connect after this update unless `"bareElicitationCapability": true` is added to their MCP config entry.

! Opus 4.7+ and Fable now default to a 1M context window on Bedrock, Vertex, Foundry, and the Claude apps gateway (the `[1m]` suffix is removed); set `CLAUDE_CODE_DISABLE_1M_CONTEXT=1` to restore the 200K window.

WAS THIS USEFUL? 👍 👎

GitHub Copilot CLI

Feedback ⓘ v1.0.91 NOTES ↗ RANK

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

> GitHub Copilot CLI added computer-use support for interacting with desktop GUIs on macOS and Windows, a new `copilot sandbox ca` command suite for proxy CA trust management that replaces the old install command, and several sandbox execution improvements including automatic review of read-only pipelines and Windows-specific fixes.

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

01 `copilot sandbox ca` command suite for proxy CA trust

90

BREAKING

Adds `copilot sandbox ca` subcommands — `check`, `create`, `trust`, `rotate`, and `remove` — to manage proxy CA trust, including unattended Windows setup. This replaces the former `/sandbox ca install` command, splitting it into separate `create` and `trust` actions; scripts calling `/sandbox ca install` must be updated.

Set up and trust a proxy CA certificate in a corporate environment where Copilot sandbox traffic passes through a TLS-intercepting proxy.

```
$ copilot sandbox ca create && copilot sandbox ca trust
```

Rotate an existing proxy CA certificate after it has been renewed, without tearing down the full sandbox configuration.

```
$ copilot sandbox ca rotate
```

— Names every subcommand with runnable example commands and migration note. v1.0.91

THINNER COVERAGE BELOW

02 **Automatic review for read-only shell pipelines** NEW

50

Allows complete, statically analyzable read-only shell pipelines to enter execution-evidence review automatically, while incomplete or unbound pipelines still require explicit approval.

— Describes the behavior distinction but no command to invoke it. v1.0.91

03 **Windows sandbox execution fixes** IMPROVED

45

Extends sandboxed command execution to Windows versions that lack filesystem enumeration support, with a warning that PowerShell's current location may be wrong, and offers a sandbox network bypass for Node/npm EACCES socket denials on Windows.

— Names concrete Windows issues fixed but no configuration steps given. v1.0.91

04 **Computer use support for desktop GUIs** NEW

40

Adds computer use capability in local sessions on macOS and Windows, letting Copilot interact with desktop applications — including legacy and GUI-only software that lacks an API, CLI, or MCP integration.

— Explains capability and scope but gives no command or setup path. product docs

└──▷ BREAKING ON UPGRADE

! The `/sandbox ca install` command is replaced by two separate commands: `create` and `trust` — any scripts or workflows calling `/sandbox ca install` must be updated.

v1.0.91 NOTES ↗ LATEST STABLE

Adds `copilot sandbox ca` command suite for proxy CA trust management and unlocks read-only pipeline execution-evidence review.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.0.91 https://github.com/github/copilot-cli.git
# already have the repo? check out this version:
$ git checkout v1.0.91
```

└──▷ TRY IT

Set up and trust a proxy CA certificate in a corporate environment where Copilot sandbox traffic passes through a TLS-intercepting proxy.

```
$ copilot sandbox ca create && copilot sandbox ca trust
```

Rotate an existing proxy CA certificate after it has been renewed, without tearing down the full sandbox configuration.

```
$ copilot sandbox ca rotate
```

Adds `copilot sandbox ca` subcommands — `check`, `create`, `trust`, `rotate`, and `remove` — to manage proxy CA trust, replacing the former `/sandbox ca install` with the split `create` and `trust` actions; includes unattended Windows setup.

Allows complete, statically analyzable read-only shell pipelines to enter execution-evidence review automatically, while incomplete or unbound pipelines still require explicit approval.

Extends sandboxed command execution to Windows versions that lack filesystem enumeration support, with a warning that PowerShell's current location may be wrong.

Offers sandbox network bypass for Node/npm EACCES socket denials on Windows.

> └─▷ **BREAKING ON UPGRADE**
>
> ! The `/sandbox ca install` command is replaced by two separate commands: `create` and `trust` — any scripts or workflows calling `/sandbox ca install` must be updated.

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-09-02 CODE ↗

RANK

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

> Diagram Design added an Architecture Delta comparison mode and expanded its layout grammar library with ten new designs across light, dark, and full-editorial variants.

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

01 **Architecture Delta topology comparison** NEW 62

Adds `type-architecture-delta` (documented in `references/type-architecture-delta.md`) for comparing synchronized topologies through a Before · Changes · After ledger that tracks added, removed, changed, moved, and rewired objects.

— Names config file and mechanism but no usage example. product docs

THINNER COVERAGE BELOW

02 **Ten new layout grammars in v2.5.10** NEW 45

Adds ten new layout grammars, each available in light, dark, and full-editorial variants (30 variants total), previewed in `.github/pr-previews/editorial-diagrams-2.5.10.jpg`.

— Gives counts and preview path but not grammar names. product docs

changes since 2026-09-02 CODE ↗

Diagram Design: Adds Marimekko as a Treemap variant.

└──▷ **GET THIS VERSION**

```
$ git clone --branch commits-2026-09-02
https://github.com/cathrynlavery/diagram-design.git
# already have the repo? check out this version:
$ git checkout commits-2026-09-02
```

Adds Streamgraph as a Line variant.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

PydanticAI

Feedback ⓘ v2.53.0 NOTES ↗ RANK

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

> PydanticAI v2.53.0 ships ToolCallJudge for pre-execution tool call review, SystemOneModel for running decision models over `/v1/systemone`, AbsurdDurability as a harness replacement for pydantic-ai-absurd, and a broad set of CLAI2 plugin, CLI and telemetry additions, alongside breaking changes to the logfire plugin name and ConcurrencyLimiter semantics.

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

01 **SystemOneModel decision model runner** NEW 65

Adds `SystemOneModel` to run decision models such as CLM and Laya over the `/v1/systemone` API.

— Names the API endpoint and supported models. v2.53.0

02 **CLAI2 plugin system enhancements** NEW 65

Adds an opt-in `herdr` plugin for `clai2`, lets `clai2` plugins run models under their own prefix with `host.model_provider`, makes `clai2` plugins declarative `Plugin` subclasses modeled on `AbstractCapability`, and tags `clai2` plugin settings with the features they need, skipping a plugin capability that rejects its settings at run setup.

— Names config keys and base class but no code sample. v2.53.0

03 **CLAI2 CLI commands and subagents** NEW 65

Adds a `/fast` command (Codex-only) to CLAI2, lets `/login` take a provider name with plugins able to add their own sign-ins, adds an `/update` command with stable (PyPI) and bleeding (`main`) channels, and adds managed subagents to CLAI2 with Claude and Codex agent definitions in Harness.

— Names exact commands and channels, no example transcript. v2.53.0

04 **ConcurrencyLimiter sharing and acquisition changes** BREAKING 60

A model wrapper now raises `UserError` when it shares a `ConcurrencyLimiter` with the agent making the request or with an enclosing model wrapper, and ConcurrencyLimiter.acquire() now takes a slot on every call, even on the same task, so custom `AbstractConcurrencyLimiter` implementations must allow release() from another task.

— Gives exact migration requirement for custom limiter code. v2.53.0

THINNER COVERAGE BELOW

05 **Observability plugin rename and Logfire telemetry** BREAKING 55

Adds opt-in Logfire UI telemetry and a Logfire setup menu (region, sign-in, project) to `pydantic-clai2`, and renames the CLAI `logfire` plugin to `observability`; code or configuration referencing the old plugin name must be updated.

— Clear rename and menu fields, but no migration command. v2.53.0

06 **ToolCallJudge for tool call review** NEW 40

Adds `ToolCallJudge` to assess tool calls before execution.

— Named class only, no mechanism or usage shown. v2.53.0

07 **AbsurdDurability harness replacement** NEW 40

Adds `AbsurdDurability` to the harness as a replacement for `pydantic-ai-absurd`.

— Names the predecessor package but no mechanism. v2.53.0

08 **oneOf schema support in TestModel** IMPROVED 40

Adds `oneOf` schema support to `TestModel`'s generated data.

— Names the schema keyword but no example. v2.53.0

09 **GPT-Live tool result media forwarding** IMPROVED 40

Sends images and documents from GPT-Live tool results to the delegated backend.

— Describes data flow but no config or API detail. v2.53.0

10 **AskUser deferral with timeout** IMPROVED 35

Lets `AskUser` defer questions to the host and time out a slow answerer.

— Describes behaviour but no config or default timeout. v2.53.0

11 **Google grounding search usage accounting** IMPROVED 25

Counts Google grounding web search queries in `usage`.

— One-line accounting change with no further detail. v2.53.0

12 **repair_messages pipeline** NEW 15

v2.53.0 introduces a `repair_messages` pipeline, mentioned in the release summary but not further detailed in the changelog bullets.

— Named only in summary, no behaviour described. v2.53.0

└──▷ **BREAKING ON UPGRADE**

! A model wrapper now raises `UserError` when it shares a `ConcurrencyLimiter` with the agent making the request or with an enclosing model wrapper.

! ConcurrencyLimiter.acquire() now takes a slot on every call, even on the same task — custom `AbstractConcurrencyLimiter` implementations must allow release() from another task.

! The CLAI `logfire` plugin is renamed to `observability`; any code or configuration referencing the old plugin name must be updated.

v2.53.0 NOTES ↗ LATEST STABLE

PydanticAI v2.53.0 adds ToolCallJudge, SystemOneModel, repair_messages pipeline, AbsurdDurability, and new clai2 plugins for PostHog, Grain, Linear, Logfire, and Herdr.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v2.53.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.53.0
```

Adds `ToolCallJudge` to assess tool calls before execution.

Adds `SystemOneModel` to run decision models such as CLM and Laya over the `/v1/systemone` API.

Adds `AbsurdDurability` to the harness as a replacement for `pydantic-ai-absurd`.

Adds an opt-in `herdr` plugin for `clai2`.

Lets `clai2` plugins run models under their own prefix with `host.model_provider`.

+12 more

└──▷ **BREAKING ON UPGRADE**

! A model wrapper now raises `UserError` when it shares a `ConcurrencyLimiter` with the agent making the request or with an enclosing model wrapper.

! ConcurrencyLimiter.acquire() now takes a slot on every call, even on the same task — custom `AbstractConcurrencyLimiter` implementations must allow release() from another task.

! The CLAI `logfire` plugin is renamed to `observability`; any code or configuration referencing the old plugin name must be updated.

WAS THIS USEFUL? 👍 👎

OpenAI Agents SDK

Feedback ⓘ v0.23.0 NOTES ↗ RANK

OpenAI Agents SDK is an open-source framework that provides tools, handoffs, guardrails, and tracing for agentic applications.

> OpenAI Agents SDK v0.23.0 adds new controls for sandbox agents, MCP tool enumeration, and encrypted session history, focused on cost and safety tuning for long-running agents.

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

01 **Docker removal protection and memory consolidation for sandbox agents**

NEW 45

Sandbox agents gain opt-in Docker removal protection to prevent accidental container deletion, plus configurable memory consolidation turns that let you tune how often long-running agents compress their context.

– No exact config key or flag names given for either control v0.23.0

02 **Configurable MCP listing page limits** NEW 30

Adds configurable page limits to control how many pages the SDK fetches when enumerating MCP tools/resources.

– No exact parameter name or default value provided v0.23.0

03 **Encrypted session history scan budget** NEW 30

Adds an opt-in encrypted history scan budget for sessions, enabling cost control on encrypted conversation history lookups.

– No exact parameter name or limit value given v0.23.0

v0.23.0 NOTES ↗ LATEST STABLE

OpenAI Agents SDK v0.23.0 adds MCP listing page limits, sandbox Docker removal protection, configurable memory consolidation turns, and encrypted session history scan budgets

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.23.0 https://github.com/openai/openai-agents-python.git
# already have the repo? check out this version:
$ git checkout v0.23.0
```

> Adds configurable MCP listing page limits to control how many pages the SDK fetches when enumerating MCP tools/resources.
>
> Adds opt-in Docker removal protection for sandbox agents to prevent accidental container deletion.
>
> Adds configurable memory consolidation turns for sandbox agents, letting you tune how often long-running agents compress their context.
>
> Adds an opt-in encrypted history scan budget for sessions, enabling cost control on encrypted conversation history lookups.

WAS THIS USEFUL? 👍 👎

Google ADK

Feedback ⓘ v2.11.0 NOTES ↗ RANK

The Agent Development Kit is Google's code-first Python toolkit for building, evaluating, and deploying agents, with built-in tools, sessions, and deployment to Vertex AI.

> Google ADK's v2.11.0 release headlines with graceful run cancellation via `abort_signal`, a new `ModelConsultTool`, SQLite-backed agent memory, and opt-in MCP SDK 2.x support, alongside new Pydantic/list schema types for LLM agents and a vector-search Files Retrieval Tool.

└──▷ **WHAT SHIPPED · 15 FEATURES** most completely described first ? what's the number?

01 **Graceful run cancellation via abort_signal** NEW 86

Adds an `abort_signal` parameter to Runner, Workflow, and workflow nodes for graceful run cancellation; the `/run_sse` endpoint now cancels the run automatically when the client disconnects.

Stop a long-running agent run gracefully when a cancellation signal is received — for example, in a CLI tool or custom server that needs to clean up on user interrupt.

python

```python
import asyncio
from google.adk import Runner

abort_signal = asyncio.Event()

async def run_with_cancel():
    runner = Runner(agent=root_agent, abort_signal=abort_signal)
    # To cancel from anywhere:
    abort_signal.set()
```

— Named parameter, endpoint, and a runnable cancellation example v2.11.0

02 **Dev UI logo configuration flags replace runtime-config.json**

83

BREAKING

The web server now serves `/dev-ui/assets/config/runtime-config.json` per request and no longer writes `runtime-config.json` into the installed package. New `--logo-text` and `--logo-image-url` server flags configure the Dev UI logo; callers must fetch the config from the server rather than reading it from disk.

— Exact flags, endpoint path, and migration note provided v2.11.0

03 **Pydantic and list schema types for LLM agents** NEW

75

LLM agents in Python can now use Pydantic model classes (e.g. `MySchema`) as well as lists of primitives — `list[str]`, `list[int]`, `list[bool]`, `list[float]` — as schema types for agent inputs/outputs.

Use a Pydantic model to enforce structured output from an LLM agent — useful when downstream code depends on a typed response.

```python
from pydantic import BaseModel

class MySchema(BaseModel):
    name: str
    confidence: float

# Pass MySchema as the output schema for your LLM agent
agent = LlmAgent(output_schema=MySchema, ...)
```

Use a list of primitives as the schema type when an LLM agent should return a collection of simple values (e.g. extracted keywords).

```python
# Use list[str] as the output schema for keyword extraction
agent = LlmAgent(output_schema=list[str], ...)
```

— Named schema types with runnable code examples for both `product docs`

THINNER COVERAGE BELOW

04 **ModelConsultTool for inter-model consultation** NEW

55

Adds `ModelConsultTool` to let an agent consult another model mid-task, capped by per-turn and per-session budgets.

— Named tool and budget mechanism but no usage example `v2.11.0`

05 **SQLite memory service for agent memory** NEW

55

Adds a SQLite memory service selectable via a `sqlite://` memory service URI for local persistent agent memory.

— Named URI scheme gives a clear config starting point `v2.11.0`

06 **Workflow plugin system enhancements** IMPROVED

50

Adds `ToolCallIntegrityPlugin` as an optional plugin for workflow tool-call integrity checks, and plugin tool callbacks now run around `ToolNode` calls in workflows.

– Named plugin and node type but no usage shown v2.11.0

07 **User approval pause via RequestInput** IMPROVED 48

Tool nodes in workflows now pause for user approval via `RequestInput` instead of passing an error downstream.

– Named mechanism but no usage example given v2.11.0

08 **Pagination for data agent listing** IMPROVED 35

Adds pagination to `list_accessible_data_agents` in the data agent toolset.

– Named function but no parameters or defaults given v2.11.0

09 **Files Retrieval Tool with vector search** NEW 30

A new Files Retrieval Tool indexes and searches local documents using vector similarity search.

– Docs mention only capability, no config or API detail product docs

10 **BigQuery tools support CMEK environments** IMPROVED 27

BigQuery tools now run in environments where CMEK is required.

– States capability without setup detail v2.11.0

11 **MCP SDK 2.x modern-protocol support** NEW 25

Adds an opt-in modern-protocol connection path for MCP SDK 2.x servers.

– No flag or config name given to enable it v2.11.0

12 **Vertex project defaults from GCP metadata** IMPROVED 25

Defaults the Vertex project from GCP metadata when not explicitly configured.

– Simple default behavior, no config key named v2.11.0

13 **Avatar and call-limit options for server commands** NEW 23

Adds avatar configuration and max LLM call limits to server commands.

– No flag names or defaults specified v2.11.0

14 **VIDEO live sessions default avatar** IMPROVED 18

VIDEO live sessions now default to a pre-built avatar.

– No mechanism or configuration detail given v2.11.0

15 **Skill telemetry span attributes stabilized** IMPROVED

Skill telemetry span attributes promoted out of experimental status.

— No attribute names or usage detail given v2.11.0

└──▷ BREAKING ON UPGRADE

! The web server now serves `/dev-ui/assets/config/runtime-config.json` per request and no longer writes `runtime-config.json` into the installed package; use `--logo-text` and `--logo-image-url` instead of editing that file, and fetch the config from the server rather than reading it from disk.

v2.11.0 NOTES ↗ LATEST STABLE

ADK v2.11.0 adds graceful cancellation via `abort_signal`, `ModelConsultTool`, SQLite memory, and MCP SDK 2.x support.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.11.0 https://github.com/google/adk-python.git
# already have the repo? check out this version:
$ git checkout v2.11.0
```

└──▷ USE IT

Stop a long-running agent run gracefully when a cancellation signal is received — for example, in a CLI tool or custom server that needs to clean up on user interrupt.

python

```python
import asyncio
from google.adk import Runner

abort_signal = asyncio.Event()

async def run_with_cancel():
    runner = Runner(agent=root_agent, abort_signal=abort_signal)
    # To cancel from anywhere:
    abort_signal.set()
```

Adds `abort_signal` parameter to `Runner`, `Workflow`, and nodes for graceful run cancellation; `/run_sse` now cancels the run automatically when the client disconnects.

Adds `ModelConsultTool` to let an agent consult another model mid-task, capped by per-turn and per-session budgets.

Adds a SQLite memory service selectable via a `sqlite://` memory service URI for local persistent agent memory.

Adds an opt-in modern-protocol connection path for MCP SDK 2.x servers.

Adds `ToolCallIntegrityPlugin` as an optional plugin for workflow tool-call integrity checks.

+9 more

└─▷ **BREAKING ON UPGRADE**

! The web server now serves `/dev-ui/assets/config/runtime-config.json` per request and no longer writes `runtime-config.json` into the installed package; use `--logo-text` and `--logo-image-url` instead of editing that file, and fetch the config from the server rather than reading it from disk.

WAS THIS USEFUL? 👍 👎

LiveKit Agents

Feedback ⓘ livekit-agents@1.8.4 NOTES ↗

RANK

LiveKit Agents builds realtime voice and video AI agents on LiveKit's WebRTC infrastructure, with turn detection, interruption handling, and telephony.

> LiveKit Agents 1.8.4 introduces two new provider integrations — Microsoft AI as a TTS provider and the Azure Voice Live Realtime API — alongside new model support across the ElevenLabs, OpenAI and AssemblyAI plugins, a Deepgram speech-rate control, and smaller updates to SmallestAI, STT metadata capture and Baseten instruction handling.

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

01 **Speed control parameter for Deepgram TTS** NEW 80

The Deepgram TTS plugin gains a `speed` parameter to control speech rate, usable as deepgram.TTS(model='aura-2', speed=1.25).

Use the new `speed` parameter with Deepgram TTS to control speech rate in a voice agent session.

python

```python
from livekit.plugins import deepgram

session = AgentSession(
    tts=deepgram.TTS(model='aura-2', speed=1.25),
)
```

— Names the exact parameter with a runnable usage example. livekit-agents@1.8.4

02 **New model support across ElevenLabs, OpenAI and AssemblyAI plugins**

NEW 75

The ElevenLabs plugin now supports the `eleven_v4` and `eleven_v4_turbo` models for higher-quality or lower-latency TTS; the OpenAI model registry gains `gpt-5.6` models; and the AssemblyAI STT plugin now defaults to the `universal-3-6-pro` model.

Switch to ElevenLabs' new `eleven_v4` or `eleven_v4_turbo` model for higher-quality or lower-latency TTS.

```python
from livekit.plugins import elevenlabs

session = AgentSession(
    tts=elevenlabs.TTS(model='eleven_v4'),
)
```

— Names every model added, with one runnable example for ElevenLabs. `livekit-agents@1.8.4`

THINNER COVERAGE BELOW

03 **Inline mid-conversation instructions for Baseten Qwen models** NEW 40

Supports inline mid-conversation instructions for Baseten Qwen/Qwen3.8-27B models.

— Names the exact model but no usage example given. `livekit-agents@1.8.4`

04 **Microsoft AI TTS provider support** NEW 30

LiveKit Agents adds Microsoft AI as a supported text-to-speech (TTS) model provider, shipped as a new speech plugin.

— Names the provider but gives no usage, config or mechanism. `livekit-agents@1.8.4`

05 **TTS continuations protocol for SmallestAI plugin** NEW 30

Adds TTS continuations protocol support to the SmallestAI plugin.

— Names the plugin and protocol but no usage detail. `livekit-agents@1.8.4`

06 **Azure Voice Live Realtime API integration** NEW 20

LiveKit Agents adds support for the Azure Voice Live Realtime API as a new provider integration.

— Only named in the release summary; no detail or example given. `livekit-agents@1.8.4`

07 **STT event metadata recorded on user turns** IMPROVED 20

LiveKit Agents now records STT event metadata on user turns.

— States the behaviour change with no mechanism or usage. `livekit-agents@1.8.4`

> `livekit-agents@1.8.4` NOTES ↗ LATEST STABLE
>
> LiveKit Agents 1.8.4 adds Azure Voice Live Realtime API, Microsoft AI speech plugin, ElevenLabs v4 models, and Deepgram TTS speed control.

└──▷ **GET THIS VERSION**

```
$ git clone --branch livekit-agents@1.8.4
https://github.com/livekit/agents.git
# already have the repo? check out this version:
$ git checkout livekit-agents@1.8.4
```

└──▷ **USE IT**

Use the new `speed` parameter with Deepgram TTS to control speech rate in a voice agent session.

```python
from livekit.plugins import deepgram

session = AgentSession(
    tts=deepgram.TTS(model='aura-2', speed=1.25),
)
```

Switch to ElevenLabs' new `eleven_v4` or `eleven_v4_turbo` model for higher-quality or lower-latency TTS.

```python
from livekit.plugins import elevenlabs

session = AgentSession(
    tts=elevenlabs.TTS(model='eleven_v4'),
)
```

Adds `speed` parameter support to the Deepgram TTS plugin.

Supports `eleven_v4` and `eleven_v4_turbo` models in the ElevenLabs plugin.

Adds gpt-5.6 models to the OpenAI model registry.

Adds TTS continuations protocol support to the SmallestAI plugin.

Records STT event metadata on user turns.

+2 more

WAS THIS USEFUL? 👍 👎

Skyvern

Feedback ⓘ v1.0.55 NOTES ↗ RANK

Skyvern automates browser-based workflows with vision models and agents, completing form-driven tasks on sites it has not seen before.

> Skyvern shipped flexible workflow scheduling (interval and one-time runs) and a new Terminate Block, added CAPTCHA-solving in code blocks and Copilot, broadened LLM support with Claude Sonnet 5.5, GPT-6.1 Sol and Azure GPT-6 Luna, and layered on dozens of Copilot UI, ownership-tracking and resilience improvements across workflows, credentials and browser recordings.

└──▷ **WHAT SHIPPED · 40 FEATURES** most completely described first ? what's the number?

01 **Interval and one-time workflow schedules** NEW 70

Workflow schedules can now run on a fixed interval measured from an anchor time (Elapsed-Interval Schedules) instead of only fixed calendar times, or be set to run once at a specific date and time (One-Time Scheduled Runs), both configured from the Schedules panel in the Workflow editor.

Schedule a compliance workflow to run once at a specific date and time for a one-off audit window, rather than on a recurring cadence.

> 📍 In the Workflow editor, open Schedules, select One-Time Scheduled Run , and set the target date and time for the single execution.

Schedule a workflow to run once at a specific time rather than on a repeating cron — useful for one-off automation triggered at a known future moment.

> 📍 In the Skyvern dashboard, open your workflow, go to Schedules, and create a new schedule — select Run once and set the target date and time.

Run a workflow on a fixed elapsed interval from an anchor time (e.g. every 6 hours from a known start) instead of a wall-clock cron.

> 📍 In the Skyvern dashboard, open your workflow, go to Schedules, create a new schedule, select Fixed interval , set the anchor time, and specify the elapsed interval.

— Names both schedule types with concrete UI steps v1.0.55

02 **Local-storage file URI reference** NEW 60

Blocks can now read a locally uploaded file referenced by a `file://` URI, so a code or browser block can read the upload without needing an external URL.

Reference an uploaded file in a workflow block using a local-storage URI so the block can read it without an external URL.

> 📍 In a code block or browser block, set the file input value to `file://<your-local-storage-upload-id>` to have the block read the locally uploaded file.

— Exact URI scheme shown with usage example `product docs`

---

03 **New LLM model support: Claude Sonnet 5.5, GPT-6.1 Sol, Azure GPT-6 Luna**

NEW

60

Adds Claude Sonnet 5.5 LLM keys with Bedrock/Anthropic fallbacks, and defaults GPT-6 Sol to medium effort; adds GPT-6.1 Sol on OpenAI and Azure, enabled on staging and production; and enables Azure GPT-6 Luna on the API and workers.

— Names every model and provider path added `v1.0.55`

THINNER COVERAGE BELOW

04 **Terminate Block for early workflow exit** NEW

55

A new Terminate Block workflow block lets a run end early with a custom human-readable reason, instead of running through remaining blocks.

End a workflow early with a human-readable reason when a prerequisite condition is not met, so downstream blocks never run.

> 📍 In the Workflow editor, add a Terminate Block between the condition check and the next step, then set a custom reason such as Prerequisite not satisfied — aborting run.

— Clear mechanism and example, no API surface `product docs`

---

05 **Customer instructions take priority run setting** NEW

50

New run setting Customer Instructions Take Priority lets workflow instructions override Skyvern 3.0's general behavior rules, and no longer lets page content override customer-provided values.

— Clear before/after behavior change described `product docs`

---

06 **Guided template walkthrough** NEW

50

Using a home page template now copies it, opens Copilot with a numbered summary of what it does, and shows an inputs card for the credentials and values it needs.

— Describes full flow, no exact navigation path `product docs`

---

07 **Google Sheets range support in Copilot** NEW 50

Copilot now resolves a Sheets tab from a URL's `gid` at save time and shows the Range in append mode (Google Sheets Range Support).

— Names exact URL parameter used product docs

08 **Browser recording improvements in Copilot** IMPROVED 50

Browser recordings now show as a live chapter inside the Copilot chat; users can talk to Copilot while a recording is capturing; the recording sticky bar is now labeled and includes a stop control; and Record Browser no longer rejects a recording outright, instead emitting repair steps for actions it can't replay (More Resilient Browser Recordings).

— Groups four related recording changes with detail product docs

09 **Validation rejects run creation for empty workflows** BREAKING 50

Rejects run creation for workflows with no blocks with a `400` response.

— Exact status code named, behavior change for callers v1.0.55

10 **CAPTCHA solving in code blocks and Copilot** NEW 45

Code blocks and Copilot browser-code cells can now solve image-text CAPTCHAs directly, including while Copilot is scouting a page and in saved code blocks.

— Names surfaces, lacks mechanism detail v1.0.55

11 **Declared code block return schema** NEW 45

Code blocks can now declare a `return schema` for Copilot, and AI-fallback output is recorded as the block's output.

— Names the config concept, no example product docs

12 **Copilot run history filtered by status** NEW 45

Copilot can now list a workflow's recent runs filtered by status (e.g., failed or terminated), instead of only showing the most recent run.

— Specific filter behavior, no exact command product docs

13 **Creator and ownership metadata across credentials, sessions, runs, agents**

NEW 45

Credentials and browser sessions now show who created them; the agents list shows each agent's original creator and first-version creation date; and workflow runs now

record who started each run, shown on the Runs page.

— Enumerates all four objects tracked v1.0.55

14 **Copilot test results as cards** IMPROVED 45

Each block Copilot tests now shows as one card with its status, code change, and failing line.

— Names card contents, no navigation path product docs

15 **Docked Copilot composer prompts** IMPROVED 45

Pending Copilot questions now dock above the composer instead of scrolling out of view (Pinned Copilot Questions), and Copilot's sign-in and Google account requests now dock above the chat input instead of scrolling out of view (Docked Copilot Sign-In Requests).

— Groups two docking UI changes, both named product docs

16 **Copilot file input handling via URL and upload** NEW 45

Copilot now knows a file input can accept a user-supplied URL, not just an upload (File Input URLs for Copilot), and can upload a chat-attached file directly into a page's file picker.

— Names both input paths, no exact UI steps v1.0.55

17 **Copilot pagination of long run results** IMPROVED 45

Lets Copilot page a long run's results instead of losing rows past the 50k cut.

— Names exact 50k row cutoff fixed v1.0.55

18 **Clipboard paste in live browser control** NEW 45

Paste clipboard content while in control of a live browser, with one unified control bar for CDP and VNC.

— Names both control protocols, no exact steps v1.0.55

19 **Default block engine option** NEW 40

The block engine picker gains a Default engine option, separate from Skyvern 1.0 and 2.0 (now labeled Legacy).

— Names exact options, no deeper mechanism product docs

20 **Layout-resilient cached code** IMPROVED 40

Saved code blocks can now target metric cards with a label-anchored selector, so they survive page layout changes (Layout-Resilient Cached Code).

— Names selector mechanism, no example usage product docs

21 **Faster credential failure detection** IMPROVED 40

A login block now fails fast with a clear reason when it isn't bound to any of the workflow's credentials, instead of opening a browser first.

– Clear before/after behavior described product docs

22 **Full YAML workflow editing** IMPROVED 35

The workflow editor's YAML view now exposes and accepts every workflow setting, rather than a subset.

– States scope but no field list product docs

23 **Browser profile preview in workflow editor** NEW 35

The workflow editor now shows a read-only preview of the browser profile a run would load.

– Names UI location but little mechanism product docs

24 **Extended runtime for in-progress long forms** IMPROVED 35

Task runs now get more time to finish long forms that are still making visible progress, instead of stopping once the token budget runs out.

– Describes behavior change, no numeric limit given product docs

25 **Clearer Copilot change review** IMPROVED 35

Copilot now shows its proposed changes on one annotated canvas, with plans and credential receipts shown where they happened in the turn (Clearer Copilot Change Review).

– Describes layout change, no navigation given product docs

26 **Upstream output reuse in Copilot repair tests** IMPROVED 35

Testing a repaired block that reads an earlier block's output now reuses that output automatically.

– Describes mechanism, no example run product docs

27 **Secure CodeBlock file downloads** NEW 35

Lets secure CodeBlocks publish generated files as run downloads.

– Clear capability, no file format or limit given v1.0.55

28 **Signed S3 links for DataDragon reports** NEW 35

Serves DataDragon report files in the dashboard through signed S3 links.

— Names delivery mechanism, no access detail v1.0.55

29 **Copilot-managed recurring schedules** NEW 30

Copilot can now create and manage the open workflow's recurring schedules directly from chat (Copilot-Managed Schedules).

— Describes capability without a concrete example product docs

30 **Copilot page-challenge recovery while scouting** NEW 30

Copilot can now solve a page challenge or start a fresh browser when scouting a page (Copilot Page-Challenge Recovery).

— Brief description, no trigger detail product docs

31 **Browser control permission simplified** IMPROVED 30

Browser control no longer requires the User Scripts permission to be enabled.

— Clear before/after, minimal detail product docs

32 **Chronological Copilot activity log** IMPROVED 30

Copilot's activity log now reads in the order things happened, with redesigned answered-question cards (Chronological Copilot Activity Log).

— Describes reordering, no further mechanism product docs

33 **Email notification controls and onboarding sequences** NEW 30

Users can now unsubscribe from categories of email Skyvern sends (e.g., analytics digest) with a one-click link; Skyvern also now sends onboarding emails that branch on what new users did, plus an out-of-credits email.

— Names email categories, no config surface v1.0.55

34 **Web search via search API** IMPROVED 30

Searches the web through the search API instead of a browser tab.

— States mechanism switch, no API name given v1.0.55

35 **Copilot reads large pages after saved-login runs** IMPROVED 30

Lets Copilot read large pages after a saved-login run and indicates when a result is withheld.

— Describes behavior, no size threshold given v1.0.55

36 **SMS two-factor authentication for credentials** NEW

25

Adds SMS two-factor authentication for credentials, with managed phone numbers.

— Thin description, no setup detail given product docs

37 **Flexible Copilot answers** IMPROVED 25

Copilot questions can now be answered in your own words instead of only the offered choices.

— Brief, no example of free-text answer product docs

38 **Code block goal editing** IMPROVED 25

Editing a code block's Goal now keeps your text and offers to update the block to match.

— Minimal description of edit flow product docs

39 **Copilot build tests in saved browser profile** IMPROVED 25

Starts Copilot build tests in the saved browser profile.

— One-line change with little context v1.0.55

40 **More resilient Workflow Studio** IMPROVED 20

Workflow Studio now contains pane crashes instead of taking down the editor.

— Very brief stability note product docs

v1.0.55 NOTES ↗ LATEST STABLE

Skyvern v1.0.55 adds interval-based workflow schedules, one-time schedule runs, Copilot file upload, CodeBlock file downloads, and Claude Sonnet 5.5 / GPT-6.1 Sol LLM support

└──▷ GET THIS VERSION

```
$ git clone --branch v1.0.55 https://github.com/Skyvern-AI/skyvern.git
# already have the repo? check out this version:
$ git checkout v1.0.55
```

└──▷ HOW TO FIND IT

Schedule a workflow to run once at a specific time rather than on a repeating cron — useful for one-off automation triggered at a known future moment.

📍 In the Skyvern dashboard, open your workflow, go to Schedules, and create a new

schedule — select `Run once` and set the target date and time.

Run a workflow on a fixed elapsed interval from an anchor time (e.g. every 6 hours from a known start) instead of a wall-clock cron.

📍 In the Skyvern dashboard, open your workflow, go to Schedules, create a new schedule, select `Fixed interval`, set the anchor time, and specify the elapsed interval.

Adds Claude Sonnet 5.5 LLM keys with Bedrock/Anthropic fallbacks, and defaults GPT-6 Sol to medium effort.

Adds GPT-6.1 Sol on OpenAI and Azure, and enables Azure on staging and production.

Enables Azure GPT-6 Luna on the API and workers.

Lets a workflow schedule run once at a set time.

Lets secure CodeBlocks publish generated files as run downloads.

+12 more

WAS THIS USEFUL? 👍 👎

EXXETA exxperts

Feedback ⓘ v0.14.1 NOTES ↗ RANK

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

> exxperts v0.14.1 expands model picker support across Claude, ChatGPT, and Copilot, and dramatically slims down the desktop app's install footprint.

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

01 **Expanded model picker across Claude, ChatGPT, Copilot** NEW

60

Adds Claude Sonnet 5.5 to the Claude model picker with variable thinking levels from low to max, adds GPT-6.1 Sol to the ChatGPT model picker, and expands the Copilot model catalogue to include Opus 5.5, Sonnet 5.5, and the GPT-6 line.

— Names specific models and pickers but no mechanism beyond selection. v0.14.1

THINNER COVERAGE BELOW

02 **Leaner desktop install with no compile step** IMPROVED

55

The desktop app now ships ~2,200 files instead of ~32,500 and starts its server without a compile step, dramatically reducing install size and startup time.

— Concrete before/after numbers but no actionable steps for user. v0.14.1

v0.14.1 NOTES ↗ LATEST STABLE

exxperts v0.14.1 adds Claude Sonnet 5.5, GPT-6.1 Sol, and a dramatically leaner desktop install (~2,200 files vs 32,500).

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.14.1 https://github.com/EXXETA/exxperts.git
# already have the repo? check out this version:
$ git checkout v0.14.1
```

Adds Claude Sonnet 5.5 to the Claude model picker, supporting variable thinking levels from low to max.

Adds GPT-6.1 Sol to the ChatGPT model picker.

Expands the Copilot model catalogue to include Opus 5.5, Sonnet 5.5, and the GPT-6 line.

> Desktop app now ships ~2,200 files instead of ~32,500 and starts its server without a compile step, dramatically reducing install size and startup time.

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ python/v1.57.2 NOTES ↗

RANK

Strands Agents is AWS's open-source SDK for production agents in Python and TypeScript, with a model-driven loop, tool integration, and deployment targets.

> Strands Agents v1.57.2 graduates its bidirectional streaming API to stable and polishes the CLI with a new `strands update` subcommand, while advancing TypeScript SDK parity with Python.

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

01 `strands update` CLI subcommand and flow polish NEW

66

Adds a `strands update` subcommand to the CLI for upgrading the Strands CLI and related packages in one step, alongside polish to the `compact`, `skills`, `rename`, and `setup` flows.

Upgrade the Strands CLI and all related packages to the latest version in one step.

```
$ strands update
```

— Exact subcommand given with a runnable example python/v1.57.2

THINNER COVERAGE BELOW

02 Bidirectional streaming API (BidiAgent) graduates to stable

IMPROVED

58

The bidirectional streaming API is promoted from experimental to stable, including `BidiAgent` agent-level storage, grouped content messages, grouped tool-call preservation through execution and delivery, text and reasoning stream events, and shared-console typing/speech support. Tool calls are now also displayed in console I/O during bidirectional streaming sessions.

— Names mechanism and sub-features but no command or config shown python/v1.57.2

03 TypeScript SDK parity improvements NEW

40

The TypeScript SDK gains ported and parity features: the `a2a_client` vended tool is ported from Python, `LocalAgent` is brought to parity with the TypeScript interface, and `InterruptError` is now exported from the TypeScript SDK.

— Named surfaces but each described in one line only python/v1.57.2

python/v1.57.2 NOTES ↗ LATEST STABLE

Strands Agents v1.57.2 graduates bidirectional streaming, adds cancellation from custom tools, and polishes the CLI with `strands update`.

└──▷ **GET THIS VERSION**

```
$ git clone --branch python/v1.57.2 https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout python/v1.57.2
```

└──▷ **TRY IT**

Upgrade the Strands CLI and all related packages to the latest version in one step.

```
$ strands update
```

Graduates the bidirectional streaming API (`BidiAgent`) to stable, including agent-level storage, grouped content messages, grouped tool-call preservation through execution and delivery, text and reasoning stream events, and shared-console typing/speech support.

Displays tool calls in console I/O during bidirectional streaming sessions.

Adds `strands update` subcommand to the CLI, alongside polish to `compact` , `skills` , `rename` , and `setup` flows.

Ports the `a2a_client` vended tool to TypeScript.

Brings `LocalAgent` to parity with the TypeScript interface.

+1 more

WAS THIS USEFUL? 👍 👎

Microsoft Agent Framework

Feedback ⓘ python-1.20.0

NOTES ↗ RANK

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

> Microsoft Agent Framework's python-1.20.0 release redesigns Foundry hosting around durable, sandbox-isolated sessions, adds native computer-use support and new DuckDB, SQL Server and TypeSafe connectors, and extends Gemini, Bedrock, orchestrations and AG-UI with a string of smaller capabilities, alongside breaking changes to Gemini's settings namespace and experimental declarative workflows.

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

01 Parameterized knowledge-source queries in Azure AI Search context provider

NEW

90

Adds `knowledge_source_params` support to `AzureAISearchContextProvider`, allowing parameterized knowledge-source queries such as `queryType` and `semanticConfiguration` to be passed when constructing the provider.

Pass parameterized knowledge-source options when building an Azure AI Search context provider.

python

```python
from agent_framework.azure_ai_search import
AzureAISearchContextProvider

provider = AzureAISearchContextProvider(
    endpoint="https://<your-search>.search.windows.net",
    index_name="my-index",
    credential=AzureCliCredential(),
    knowledge_source_params={"queryType": "semantic",
"semanticConfiguration": "my-config"},
)
```

— Full runnable code example shows exact parameter usage. python-1.20.0

02 Foundry hosting redesign for native workflows

BREAKING

80

Adds `agent-framework-foundry-hosting` support for hosting native workflows over Foundry Responses and Foundry Invocations, then redesigns it around request-scoped agent factories, persistent sandbox-isolated sessions, parsed and durable Invocations runs, and configurable Responses history, background execution, and options. Existing hosting setups must be migrated to the new model.

— Names redesign scope in detail but gives no migration steps. python-1.20.0

---

03 **Gemini Embedding 2 and GOOGLE_* settings namespace** BREAKING

70

Adds Gemini Embedding 2 and vector task options to `agent-framework-gemini`, switches the provider to the `GOOGLE_*` settings namespace, and supports operation-specific embedding options in shared vector APIs. Existing configuration using a different namespace must be updated to `GOOGLE_*` keys.

— States breaking namespace change but gives no exact key mapping. python-1.20.0

---

04 **AG-UI durable disconnected runs and approval isolation**

IMPROVED

65

Makes disconnected AG-UI runs durable, emits `RUN_STARTED` before execution when IDs are supplied, and isolates custom approval-state namespaces in `agent-framework-ag-ui`.

— Behavior changes named precisely but no usage example. python-1.20.0

---

05 **DuckDB and SQL Server vector-store connectors** NEW

60

Adds `agent-framework-duckdb` and `agent-framework-sql-server` native vector-store connectors, providing native vector search integrations for DuckDB and SQL Server.

Use the new DuckDB native vector-store connector for fast local vector search in an agent pipeline.

```
$ pip install agent-framework-duckdb
```

— Install command given but no usage beyond that. python-1.20.0

---

06 **Magentic orchestration controls** NEW

60

Adds per-workflow Magentic manager creation, optional disable of return-to-previous handoff routing, approval-gated sequential output, and generated-handoff validation to `agent-framework-orchestrations`.

— Four orchestration additions named but no usage example. python-1.20.0

07 **New core primitives: injection context, checkpoint typing, stream buffering, file isolation**

NEW 60

Adds `FunctionInvocationContext` opt-in injection for skill scripts, an `AgentExecutor` checkpoint-state `TypedDict` for typed checkpoint inspection and manipulation, response-stream gates and buffering, and session-scoped file-access isolation to `agent-framework-core`.

— Four named core surfaces but no usage examples. `python-1.20.0`

THINNER COVERAGE BELOW

08 **Bedrock streaming, image sending and guardrail forwarding**

NEW 55

Adds streaming support via `ConverseStream`, user-image sending, and Converse guardrail/additional-model-request option forwarding to `agent-framework-bedrock`.

— Named API surfaces but no usage example. `python-1.20.0`

09 **Native computer-use support for Responses and Foundry embeddings**

NEW 55

Adds native computer-use support to Responses clients and Foundry project embeddings via `agent-framework-foundry`, `agent-framework-foundry-hosting`, and `agent-framework-openai`.

— Packages named but mechanism and usage unexplained. `python-1.20.0`

10 **Behavior changes in experimental declarative workflows**

55

BREAKING

In the experimental `agent-framework-declarative` package, workflow state attribute lookup, bound PowerFx state construction, workflow HTTP request URL normalization, and `SendActivity` expression-result preservation have changed, so existing declarative workflow definitions may behave differently.

— Four changed behaviors named but no migration guidance. `python-1.20.0`

11 **TypeSafe AI connector** NEW 50

Adds the `agent-framework-typesafe` TypeSafe AI connector, a new strongly-typed AI provider integration.

Use the TypeSafe AI connector to add a strongly-typed AI provider to your agent.

```
$ pip install agent-framework-typesafe
```

– Install command given but no further usage detail. python-1.20.0

12 **Structured input and image conversion in ChatKit** NEW 35

Adds support for structured input and generated-image conversion in `agent-framework-chatkit`.

– Single-line description with no example or mechanism. python-1.20.0

13 **MongoDB agent RAG sample** NEW 30

Adds a MongoDB agent RAG sample via `agent-framework-mongodb`.

– Only a sample is mentioned, no further detail. python-1.20.0

└──▷ **BREAKING ON UPGRADE**

! `agent-framework-foundry-hosting` has been redesigned around request-scoped agent factories, persistent sandbox-isolated sessions, parsed and durable Invocations runs, and configurable Responses history, background execution, and options — existing hosting setups must be migrated.

! `agent-framework-declarative` (experimental): workflow state attribute lookup, bound PowerFx state construction, workflow HTTP request URL normalization, and `SendActivity` expression-result preservation have changed — existing declarative workflow definitions may behave differently.

! `agent-framework-gemini` now uses the `GOOGLE_*` settings namespace — any configuration previously using a different namespace must be updated to `GOOGLE_*` keys.

python-1.20.0 NOTES ↗ LATEST STABLE

Adds DuckDB/SQL Server vector-store connectors, TypeSafe AI connector, Foundry Responses/Invocations hosting, native computer-use, and new core primitives.

└──▷ **GET THIS VERSION**

```
$ git clone --branch python-1.20.0
https://github.com/microsoft/agent-framework.git
# already have the repo? check out this version:
$ git checkout python-1.20.0
```

└──▷ **TRY IT**

Use the new DuckDB native vector-store connector for fast local vector search in an agent pipeline.

```
$ pip install agent-framework-duckdb
```

Use the TypeSafe AI connector to add a strongly-typed AI provider to your agent.

```
$ pip install agent-framework-typesafe
```

Pass parameterized knowledge-source options when building an Azure AI Search context provider.

```python
from agent_framework.azure_ai_search import
AzureAISearchContextProvider

provider = AzureAISearchContextProvider(
    endpoint="https://<your-search>.search.windows.net",
    index_name="my-index",
    credential=AzureCliCredential(),
    knowledge_source_params={"queryType": "semantic",
"semanticConfiguration": "my-config"},
)
```

Adds `knowledge_source_params` support to `AzureAISearchContextProvider` for parameterized knowledge-source queries.

Adds `FunctionInvocationContext` opt-in injection for skill scripts in `agent-framework-core`.

Adds `AgentExecutor` checkpoint-state `TypedDict` for typed checkpoint inspection and manipulation.

Adds response-stream gates and buffering to `agent-framework-core`.

Adds session-scoped file-access isolation to `agent-framework-core`.

+11 more

└──▷ **BREAKING ON UPGRADE**

! `agent-framework-foundry-hosting` has been redesigned around request-scoped agent factories, persistent sandbox-isolated sessions, parsed and durable Invocations runs, and configurable Responses history, background execution, and options — existing hosting setups must be migrated.

! `agent-framework-declarative` (experimental): workflow state attribute lookup, bound PowerFx state construction, workflow HTTP request URL

normalization, and `SendActivity` expression-result preservation have changed — existing declarative workflow definitions may behave differently.

! `agent-framework-gemini` now uses the `GOOGLE_*` settings namespace — any configuration previously using a different namespace must be updated to `GOOGLE_*` keys.

WAS THIS USEFUL? 👍 👎

OpenClaw

Feedback ⓘ v2026.8.35 NOTES ↗ CODE ↗ RANK

OpenClaw is a personal AI assistant platform that connects messaging channels, tools, and model providers through a gateway.

Extended-stable maintenance release with reliability, channel, agent, secrets, and plugin-state fixes across Gateway, cron, WebChat, and llama.cpp.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v2026.8.35
https://github.com/openclaw/openclaw.git
# already have the repo? check out this version:
$ git checkout v2026.8.35
```

NEW INTEGRATION GPT-6.1 Sol model support

WAS THIS USEFUL? 👍 👎

Agno (formerly Phidata)

Feedback ⓘ v3.1.1 NOTES ↗

RANK

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

> Agno v3.1.1 adds real-time progress tracking and cancellation for knowledge page syncs, plus native progress events for workflow steps streamed to AgentOS clients.

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

01 Knowledge page sync progress and cancellation NEW

95

Adds Knowledge.stream_sync_pages() and astream_sync_pages() methods that yield typed `PageSyncProgress` snapshots and a terminal `SyncReport`, enabling real-time progress tracking during knowledge page synchronisation. Also adds an `on_progress` observer parameter to `sync_pages` and `async_sync_pages` for callback-based monitoring without switching to a streaming call, and a `SyncReport.failed_paths` field listing site paths of pages that failed to publish or delete (capped at 20, matching the existing `errors` cap; `failed` retains the full count; defaults to ()). Cancelling a page sync now propagates to the in-progress work instead of draining the remainder of the run.

Stream live progress snapshots while syncing a large knowledge corpus, so you can surface per-page status to users or logs without polling.

python

```
async for event in knowledge.astream_sync_pages():
    if isinstance(event, PageSyncProgress):
        print(event)
    else:  # terminal SyncReport
        print('failed pages:', event.failed_paths)
```

Inspect which pages failed during a sync run when using the non-streaming path, without changing existing call sites.

python

```
report = knowledge.sync_pages(
    on_progress=lambda p: print(p)
)
print('Failed paths:', report.failed_paths)
print('Total failures:', report.failed)
```

— Names exact methods, fields, caps, with runnable code. v3.1.1

02 **Workflow step progress events over SSE** NEW 80

Workflow function steps can now yield StepProgress(content=..., data=...) before their `StepOutput`, emitted as a native `StepProgressEvent` and streamed over the existing workflow REST/SSE route. AgentOSClient.run_workflow_stream() is extended to parse these `StepProgressEvent` messages delivered over that same route.

Emit incremental progress from a workflow step so AgentOS streams it to the client over the existing SSE route before the final output.

python

```
def my_step(input) -> StepOutput:
    yield StepProgress(content='Fetching data...', data={'pct':
10})
    result = do_work(input)
    yield StepProgress(content='Processing...', data={'pct': 80})
    return StepOutput(content=result)
```

— Names the classes and method, with a runnable example. v3.1.1

v3.1.1 NOTES ↗ LATEST STABLE

Agno v3.1.1 adds streaming page-sync progress, cancellation, and named failure reporting via new methods and observer hooks.

└──▷ GET THIS VERSION

```
$ git clone --branch v3.1.1 https://github.com/agno-agi/agno.git
# already have the repo? check out this version:
$ git checkout v3.1.1
```

└──▷ USE IT

Stream live progress snapshots while syncing a large knowledge corpus, so you can surface per-page status to users or logs without polling.

python

```
async for event in knowledge.astream_sync_pages():
    if isinstance(event, PageSyncProgress):
        print(event)
    else:  # terminal SyncReport
        print('failed pages:', event.failed_paths)
```

Inspect which pages failed during a sync run when using the non-streaming path, without changing existing call sites.

```python
report = knowledge.sync_pages(
    on_progress=lambda p: print(p)
)
print('Failed paths:', report.failed_paths)
print('Total failures:', report.failed)
```

Emit incremental progress from a workflow step so AgentOS streams it to the client over the existing SSE route before the final output.

```python
def my_step(input) -> StepOutput:
    yield StepProgress(content='Fetching data...', data={'pct': 10})
    result = do_work(input)
    yield StepProgress(content='Processing...', data={'pct': 80})
    return StepOutput(content=result)
```

Adds Knowledge.stream_sync_pages() and astream_sync_pages() methods that yield typed `PageSyncProgress` snapshots and a terminal `SyncReport`, enabling real-time progress tracking during knowledge page synchronisation.

Adds `on_progress` observer parameter to `sync_pages` and `async_sync_pages` for callback-based progress monitoring without switching to a streaming call.

Adds `SyncReport.failed_paths` field listing site paths of pages that failed to publish or delete (capped at 20, matching the existing `errors` cap; `failed` retains the full count; defaults to ()).

Enables workflow function steps to yield StepProgress(content=..., data=...) before their `StepOutput`, emitted as a native `StepProgressEvent` and streamed over the existing workflow REST/SSE route.

Extends AgentOSClient.run_workflow_stream() to parse `StepProgressEvent` messages delivered over the existing workflow SSE route.

+1 more

WAS THIS USEFUL? 👍 👎

Cloudflare Agents

Feedback ⓘ @cloudflare/ai-chat@0.12.1

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 shipped significant Browser Run/CDP tooling, new experimental model providers and a pi harness, session-mirroring and agent-tool event controls, plus several breaking changes around scheduled-task scoping, Durable Object ID addressing, and AI SDK dependency requirements.

└──▷ **WHAT SHIPPED · 15 FEATURES** most completely described first ? what's the number?

01 **whenChanged remind option for context blocks** NEW 85

Context blocks gain a `whenChanged: "remind"` option that sends changes as a lightweight reminder appended after the cached prompt instead of rewriting the whole prompt, reducing token spend on prompt-cache hits.

Suppress a full prompt rewrite on context change and instead append a lightweight reminder after the cached prompt, reducing token spend on prompt-cache hits.

typescript

```
// inside a Think context block definition
{ whenChanged: "remind" }
```

— Named config option with mechanism and runnable snippet @cloudflare/think@0.20.0

02 **Browser Run and CDP session improvements** NEW 85

`loadCdpSpec` now accepts an optional `sessionId` to read the CDP protocol from an existing Browser Run session, and cdp.spec()/loadCdpSpec() now include command parameters, return types, and type details so a model can see how to call a CDP method. Browser Run adds hostname guardrails, close and activity callbacks, and the `CdpConnection` name (retaining `CdpSession` as a deprecated alias), plus persistent browser tools for AI SDK and TanStack AI backed by experimental Browser sessions.

— Enumerates every named surface in the module agents@0.25.0

03 **Silent rejection via rejectExecution autoContinue** NEW 80

rejectExecution() gains an `{ autoContinue: false }` option to record a rejection without starting another agentic turn — useful when a human denies a request and the rejection should be logged silently.

Reject a pending action without triggering another agentic turn — useful when a human denies a request and you want to log the rejection silently.

```typescript
await rejectExecution({ autoContinue: false });
```

— Named option with runnable example and use case `@cloudflare/think@0.20.0`

04 Terminal-only event delivery for runAgentTool NEW

75

`runAgentTool` gains an `eventDelivery: "terminal"` option that forwards only lifecycle, progress, and milestone events from a child agent run, keeping a parent agent's event stream quiet during long sub-agent work.

Reduce noise in parent agent timelines by receiving only lifecycle, progress, and milestone events from a child agent tool run.

```typescript
const result = await runAgentTool(env, {
  agent: ChildAgent,
  input: userMessage,
  eventDelivery: 'terminal'
});
```

Receive only terminal-phase events from a child agent-tool run, keeping the parent's event stream quiet during long sub-agent work.

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

const result = await runAgentTool(childAgent, input, {
  eventDelivery: 'terminal'
});
```

— Names exact flag and shows runnable usage `agents@0.25.0` `@cloudflare/ai-chat@0.12.1`

05 session.mirror() for in-memory transcripts NEW

75

session.mirror() keeps an in-memory transcript automatically in sync with a session's change feed, enabling read-only observers or audit logs without polling or manual diffing.

Keep an in-memory transcript up to date as a session evolves, without polling or manual diffing.

```typescript
const transcript = await session.mirror();
```

Mirror a session's change feed into an in-memory transcript to build read-only observers or audit logs without polling.

```typescript
const mirror = session.mirror();
```

— Named method with clear mechanism and runnable call `@cloudflare/think@0.20.0` `@cloudflare/ai-chat@0.12.1`

06 **onTurnEnd callback in useAgentChat** NEW 75

`useAgentChat` gains an `onTurnEnd` callback that fires when a chat turn completes, and terminal chat frames now carry outcomes and user message IDs so clients can settle sends correctly across recovery and reconnects.

React to each completed turn in a chat UI — e.g. to trigger analytics or unlock UI elements after the assistant finishes responding.

```typescript
const { messages } = useAgentChat({
  agent,
  onTurnEnd: ({ outcome, userMessageId }) => {
    console.log('Turn finished:', outcome, userMessageId);
  }
});
```

— Names callback and behaviour across recovery/reconnect `@cloudflare/ai-chat@0.12.1`

07 **Scheduled tasks scoped to root agent by default** BREAKING 70

Declared scheduled tasks now run on the root agent only by default; override getScheduledTasksScope() to return `"all"` for per-facet scheduling.

— Names method and migration override `@cloudflare/think@0.20.0`

08 **reconcileMessages replaces resolveToolMergeId** DEPRECATED 65

Introduces `reconcileMessages` as the replacement for the deprecated `resolveToolMergeId`, keeping assistant messages and tool outputs attached to the correct turn when a provider reuses a `toolCallId`.

– Explains mechanism but no code example `@cloudflare/think@0.20.0` `@cloudflare/ai-chat@0.12.1`

09 **Updated AI SDK and agents dependency requirements** BREAKING

60

Think now requires `ai@^7` and `@ai-sdk/react@^4` because it bundles providers that return AI SDK v7 models, and requires `agents@>=0.25.0` so Think and AI Chat can import the chat helpers provided by that release; prior versions of these packages are no longer compatible.

– Names exact version constraints and reason `@cloudflare/think@0.20.0`

10 **Experimental pi-ai provider and pi harness packages** NEW

60

Adds `agents/models/pi-ai`, an experimental pi-ai provider for Workers AI and AI Gateway over the `AI` binding (also referenced as `agents/models/ai-sdk`), and `agents/harnesses/pi`, an experimental harness to host pi-durable sessions in a Durable Object.

– Names new experimental packages but no usage shown `agents@0.25.0`

THINNER COVERAGE BELOW

11 **Raw Durable Object IDs fail first async RPC** BREAKING

55

Raw IDs from newUniqueId() and idFromString() now fail their first async RPC — Agents should be addressed by name instead.

– Names methods and migration guidance `@cloudflare/think@0.20.0`

12 **useAgentChat reloads history on agent change** IMPROVED

55

`useAgentChat` now loads the new agent's history when the agent address passed to `useAgent` changes.

– Clear behaviour but no example `agents@0.25.0`

13 **Recovered flag on onChatResponse** IMPROVED

50

`onChatResponse` now fires with `recovered: true` for a turn that was persisted just before a Durable Object reset.

– Named field but no usage example `@cloudflare/think@0.20.0`

14 **Interfaces usable as Agent Props** IMPROVED

40

Interfaces can now be used as `Props` on `Agent`, `AIChatAgent`, `Think`, and routing helpers.

— Names surfaces but no example or mechanism `@cloudflare/ai-chat@0.12.1`

15 **Browser chat turns use web channel** IMPROVED 40

Browser chat turns now use the `web` channel, so its configured tools and turn limit apply.

— Thin description, no example `@cloudflare/think@0.20.0`

└─▷ **BREAKING ON UPGRADE**

! Declared scheduled tasks now run on the root agent only by default; override getScheduledTasksScope() to return `"all"` for per-facet scheduling.

! Raw IDs from newUniqueId() and idFromString() now fail their first async RPC — address Agents by name instead.

! `resolveToolMergeId` is deprecated in favour of `reconcileMessages`.

! Requires `ai@^7` and `@ai-sdk/react@^4`; prior versions of those packages are no longer compatible.

! Requires `agents@>=0.25.0` so Think and AI Chat can import the chat helpers provided by that release.

`@cloudflare/ai-chat@0.12.1` NOTES ↗ LATEST STABLE

Adds `eventDelivery: "terminal"` to `runAgentTool`, session.mirror(), and `onTurnEnd` to `useAgentChat` for richer agent orchestration.

└─▷ **GET THIS VERSION**

```
$ git clone --branch @cloudflare/ai-chat@0.12.1
https://github.com/cloudflare/agents.git
# already have the repo? check out this version:
$ git checkout @cloudflare/ai-chat@0.12.1
```

└─▷ **USE IT**

Reduce noise in parent agent timelines by receiving only lifecycle, progress, and milestone events from a child agent tool run.

typescript

```
const result = await runAgentTool(env, {
  agent: ChildAgent,
  input: userMessage,
  eventDelivery: 'terminal'
});
```

Keep an in-memory transcript up to date as a session evolves, without polling or manual diffing.

```typescript
const transcript = await session.mirror();
```

React to each completed turn in a chat UI — e.g. to trigger analytics or unlock UI elements after the assistant finishes responding.

```typescript
const { messages } = useAgentChat({
  agent,
  onTurnEnd: ({ outcome, userMessageId }) => {
    console.log('Turn finished:', outcome, userMessageId);
  }
});
```

Adds `eventDelivery: 'terminal'` option to `runAgentTool` to forward only lifecycle, progress, and milestone events for a run.

Adds session.mirror() to keep an in-memory transcript in sync with a session's change feed.

`useAgentChat` gains an `onTurnEnd` callback; terminal chat frames now carry outcomes and user message IDs so clients can settle sends correctly across recovery and reconnects.

Interfaces can now be used as `Props` on `Agent`, `AIChatAgent`, `Think`, and routing helpers.

Introduces `reconcileMessages` as the replacement for the deprecated `resolveToolMergeId`, keeping assistant messages and tool outputs attached to the correct turn when a provider reuses a `toolCallId`.

`@cloudflare/think@0.20.0` NOTES ↗ STABLE

Cloudflare Agents: Adds `whenChanged: "remind"` to context blocks to send changes as a reminder after the cached prompt instead of rewriting it.

└──▷ **GET THIS VERSION**

```
$ git clone --branch @cloudflare/think@0.20.0
https://github.com/cloudflare/agents.git
# already have the repo? check out this version:
$ git checkout @cloudflare/think@0.20.0
```

└──▷ **USE IT**

Mirror a session's change feed into an in-memory transcript to build read-only observers or audit logs without polling.

```typescript
const mirror = session.mirror();
```

Reject a pending action without triggering another agentic turn — useful when a human denies a request and you want to log the rejection silently.

```typescript
await rejectExecution({ autoContinue: false });
```

Suppress a full prompt rewrite on context change and instead append a lightweight reminder after the cached prompt, reducing token spend on prompt-cache hits.

```typescript
// inside a Think context block definition
{ whenChanged: "remind" }
```

Adds `{ autoContinue: false }` option to rejectExecution() to record a rejection without starting another turn.

Browser chat turns now use the `web` channel, so its configured tools and turn limit apply.

Requires `ai@^7` and `@ai-sdk/react@^4` because Think bundles providers that return AI SDK v7 models.

`onChatResponse` now fires with `recovered: true` for a turn persisted just before a Durable Object reset.

└──▷ **BREAKING ON UPGRADE**

- ! Declared scheduled tasks now run on the root agent only by default; override getScheduledTasksScope() to return `"all"` for per-facet scheduling.
- ! Raw IDs from newUniqueId() and idFromString() now fail their first async RPC — address Agents by name instead.
- ! `resolveToolMergeId` is deprecated in favour of `reconcileMessages`.
- ! Requires `ai@^7` and `@ai-sdk/react@^4`; prior versions of those packages are no longer compatible.
- ! Requires `agents@>=0.25.0` so Think and AI Chat can import the chat helpers provided by that release.

agents@0.25.0 NOTES ↗ STABLE

Cloudflare Agents: Adds `agents/models/ai-sdk`, an experimental provider for Workers AI and AI Gateway over the `AI` binding.

└──▷ **GET THIS VERSION**

```
$ git clone --branch agents@0.25.0
https://github.com/cloudflare/agents.git
# already have the repo? check out this version:
$ git checkout agents@0.25.0
```

└──▷ **USE IT**

Receive only terminal-phase events from a child agent-tool run, keeping the parent's event stream quiet during long sub-agent work.

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

const result = await runAgentTool(childAgent, input, {
  eventDelivery: 'terminal'
});
```

Adds `agents/models/pi-ai`, an experimental pi-ai provider for Workers AI and AI Gateway over the `AI` binding.

Adds `agents/harnesses/pi`, an experimental harness to host pi-durable sessions in a Durable Object.

`loadCdpSpec` accepts an optional `sessionId` to read the CDP protocol from an existing Browser Run session.

cdp.spec() and loadCdpSpec() now include command parameters, return types, and type details so a model can see how to call a CDP method.

Adds Browser Run hostname guardrails, close and activity callbacks, and the `CdpConnection` name (retaining `CdpSession` as a deprecated alias).

+2 more

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

gram

Feedback ⓘ server@2.16.0 NOTES ↗ RANK

Gram provides MCP servers and AI-agent tools from APIs with access controls and observability.

> Gram's October releases replaced Explore's saved-query service with a new `widgets` service, promoted the Access Hub to an org-wide page with live editing and automatic token exchange for identity-chained MCP servers, and built out Okta application mapping alongside a full onboarding playbook and stack/step catalog system in the admin dashboard.

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

01 Org-wide Access Hub and workloadIdentities API scoping

IMPROVED 90

The Access Hub is promoted to an org-wide page at `/<org>/access-hub`, visible in the org sidebar under Secure to any holder of `workload:read` or `workload:write`; old per-project workload identity URLs redirect there automatically. The `workloadIdentities` API no longer requires a project when called from a dashboard session, reading and writing at the organization tier; API-key callers must still name a project, and `project_scoped` still requires one.

Navigate to the org-wide Access Hub to review all workload identities across every project in one place.

> 📍 Navigate to /<org> `/access-hub` in your browser (requires `workload:read` or `workload:write` on the organization). Old per-project workload identity URLs redirect here automatically.

— Gives the exact URL path and API scoping rules. server@2.16.0 dashboard@0.130.0

02 Okta application mapping to MCP catalog entries NEW

90

Okta-connected org admins can list, dismiss, and restore MCP server suggestions derived from synced Okta applications via the catalog Okta mapping, gated by the `okta-connections` rollout. Staff can map Okta Integration Network (OIN) applications to Gram-owned MCP catalog entries via the `com.speakeasy.ai/okta` metadata namespace, with registry validation enforcing at most one entry per OIN application name and the mapping returned as public catalog data; staff editor proposals suggest Okta application names for catalog entries matched by vendor domain, title, or tenant label, surfacing unmatched observed names for manual review without applying anything automatically.

Map an Okta Integration Network application to a Gram MCP catalog entry so the association is validated and returned to registry consumers.

```json
{ "metadata": { "com.speakeasy.ai/okta": "<OIN-application-name>" } }
```

— Names the metadata key and includes a config example. `admin@0.7.0` `server@2.16.0`

03 **Onboarding playbook assignment** NEW 88

Adds onboarding use cases and playbooks in the admin dashboard, enabled via the `assign_organization_onboarding_playbook` write operation, which assigns a playbook to an organization by ID or by use case, replacing the removed preset selection editor and its Admin API. Staff create and assign shared or per-customer playbooks; a shared playbook is assigned as a copy so later edits to the template never propagate to organizations already on it. The onboarding survey auto-assigns a use case's default playbook, the customer setup wizard walks through its steps, and Admin MCP diagnostics now reports the assigned playbook instead of task visibility.

— Names the write operation and its copy-on-assign behaviour. `admin@0.7.0` `dashboard@0.130.0` `server@2.16.0`

04 **Widgets service replaces Explore saved-query service** BREAKING 85

Adds the `widgets` service with `list`, `get`, `save`, `update`, and `duplicate` operations for Explore's saved questions; each widget is validated against the catalog and chart on save and on read, and deleting another member's widget requires project write access. This replaces the removed `explore` saved-query service (`explore.listQueries`, `createQuery`, `updateQuery`, `deleteQuery`); the `query:*` audit actions and the `audit_log.query_event_v1` webhook event are retired — nothing emits them, though existing audit entries and webhook subscriptions remain valid.

— Names every added and removed operation and audit event. `server@2.16.0`

05 **Editing access entries and trusted platforms in Access Hub** NEW 78

Adds an Edit action to each machine on a platform's Access Hub page to change its `label`, `tags`, and assigned agent, and an Edit action on the platform page to change its `name`, `description`, `tags`, and `JWKS URI` (issuer URL and wildcard admission stay fixed after registration); both actions record every edit in the audit log.

— Names every editable field; access is UI-only. `server@2.16.0`

06 **Cross-App Access readiness tool and live verification** NEW

78

Adds a read-only Platform MCP tool for administrators to inspect Cross-App Access readiness for an exact project and MCP server. Readiness now reflects live identity-chaining exchange results, showing Verified, Not confirmed, or Not working states, with result changes recorded in the audit log as `okta-resource-connection:observe`.

— Names the exact audit action and result states. server@2.16.0

07 **Onboarding stack and step catalog** NEW

73

Adds an onboarding stack and step catalog in the admin dashboard: staff record an organization's vendor stack, plans, and device management on its Overview page, drawn from a support matrix catalog that now carries plans and files every product under its real vendor; the catalog is mirrored into the database at start-up. A read-only Steps page lets setup steps nest one level under a group, with each prerequisite linking to its own row; observability and MCP distribution are now step groups.

— Names the Overview and Steps pages but no API. admin@0.7.0 server@2.16.0

THINNER COVERAGE BELOW

08 **Automatic downstream token exchange for identity-chained MCP servers**

NEW

55

Remote and tunneled MCP servers with a ready identity-chaining binding now obtain downstream access tokens automatically from the signed-in user's identity provider — supporting enterprise-managed authorization and Okta Cross App Access — via identity assertion grant exchange.

— Explains the mechanism but gives no setup steps. server@2.16.0

09 **Multi-select scope picker for identity provider clients**

IMPROVED

42

Scope selection for a manually configured identity provider client is now a multi-select listing advertised scopes, with support for typing in unlisted scopes.

— Names the control but not its exact location. dashboard@0.130.0

10 **Role-based plugin distribution control** NEW

40

Adds a temporary staff-only control for enabling role-based plugin distribution per organization, with audited Admin MCP approval.

— Thin description with no named fields or steps. admin@0.7.0

11 **Findings reveal for MCP tool calls** IMPROVED

38

Findings from MCP tool calls can now be revealed; findings recorded before this release show `Evidence not stored` instead of an error.

— Brief description with no mechanism or UI path. `server@2.16.0`

└──▷ BREAKING ON UPGRADE

! The `explore` saved-query service is removed: `explore.listQueries`, `createQuery`, `updateQuery`, and `deleteQuery` no longer exist; Explore saves widgets through the `widgets` service instead.

! `query:*` audit actions and the `audit_log.query_event_v1` webhook event are retired — nothing emits them any more.

! The preset selection editor and its Admin API are removed; the Admin MCP diagnostics tool now reports the assigned playbook, and the Admin MCP's onboarding write proposal assigns a playbook by ID or use case instead of setting task visibility.

! The preset selection editor and its Admin API are removed; the Admin MCP's onboarding write proposal now assigns a playbook by ID or by use case instead of setting task visibility.

! The preset selection editor and its Admin API are removed; playbook assignment via `assign_organization_onboarding_playbook` replaces them.

`server@2.16.0` `NOTES ↗` **LATEST** `STABLE`

gram server 2.16.0 adds the `widgets` service, org-wide Access Hub, Okta MCP suggestions, enterprise identity chaining, and a read-only Platform MCP Cross-App Access tool.

└──▷ GET THIS VERSION

```
$ git clone --branch server@2.16.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout server@2.16.0
```

└──▷ HOW TO FIND IT

Navigate to the org-wide Access Hub to review all workload identities across every project in one place.

📍 Navigate to /<org>`/access-hub` in your browser (requires `workload:read` or `workload:write` on the organization). Old per-project workload identity URLs redirect here automatically.

Adds the `widgets` service with list, get, save, update, and duplicate operations for Explore's saved questions; each widget is validated against the catalog and chart on save and on read, and deletions of another member's widget require project write access.

Promotes the Access Hub to an org-wide page at `/<org>/access-hub`, visible in the org sidebar under Secure to any holder of `workload:read` or `workload:write`; old project URLs redirect there, and `workloadIdentities` API calls from a dashboard session no longer require a project.

Adds a read-only Platform MCP tool for administrators to inspect Cross-App Access readiness for an exact project and MCP server.

Enables Okta-connected org admins to list, dismiss, and restore MCP server suggestions derived from synced Okta applications via the catalog Okta mapping; gated by the `okta-connections` rollout.

Remote and tunneled MCP servers with a ready identity-chaining binding now obtain downstream access tokens automatically from the signed-in user's identity provider (enterprise-managed authorization, Okta Cross App Access) via identity assertion grant exchange.

+7 more

└──▷ **BREAKING ON UPGRADE**

! The `explore` saved-query service is removed: `explore.listQueries`, `createQuery`, `updateQuery`, and `deleteQuery` no longer exist; Explore saves widgets through the `widgets` service instead.

! `query:*` audit actions and the `audit_log.query_event_v1` webhook event are retired — nothing emits them any more.

! The preset selection editor and its Admin API are removed; the Admin MCP diagnostics tool now reports the assigned playbook, and the Admin MCP's onboarding write proposal assigns a playbook by ID or use case instead of setting task visibility.

`dashboard@0.130.0` **NOTES** ↗ **STABLE**

gram: The `workloadIdentities` API no longer requires a project when called from a dashboard session (reads/writes the organization tier); API-key callers still name a project, and `project_scoped` still requires one.

└──▷ **GET THIS VERSION**

```
$ git clone --branch dashboard@0.130.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout dashboard@0.130.0
```

Enables the `assign_organization_onboarding_playbook` write operation to assign a playbook to an organization by ID or by use case, replacing the preset selection

editor and Admin API.

Scope selection for a manually configured identity provider client is now a multi-select listing advertised scopes, with support for typing in unlisted scopes.

└──▷ **BREAKING ON UPGRADE**

! The preset selection editor and its Admin API are removed; the Admin MCP's onboarding write proposal now assigns a playbook by ID or by use case instead of setting task visibility.

admin@0.7.0 NOTES ↗ STABLE

gram admin 0.7.0 adds onboarding playbooks, stack catalog, Okta OIN mapping, and role-based plugin distribution controls.

└──▷ **GET THIS VERSION**

```
$ git clone --branch admin@0.7.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout admin@0.7.0
```

└──▷ **USE IT**

Map an Okta Integration Network application to a Gram MCP catalog entry so the association is validated and returned to registry consumers.

```json
{ "metadata": { "com.speakeasy.ai/okta": "<OIN-application-name>" } }
```

Assigns a shared playbook as a copy to the receiving organization so later edits to the shared template never propagate to organizations already on it.

Surfaces the assigned playbook in Admin MCP diagnostics and walks the customer setup wizard through its steps; the onboarding survey auto-assigns a use case's default playbook.

Adds an onboarding stack and step catalog: staff record an organization's vendors, plans, and device management on its Overview page drawn from the support matrix catalog, which now carries plans and files every product under its real vendor.

Adds a Steps page (read-only catalog mirror) in the admin dashboard where setup steps can nest one level under a group, and each prerequisite links to its own row; observability and MCP distribution become step groups.

Supports mapping Okta Integration Network applications to Gram-owned MCP catalog entries via the `com.speakeasy.ai/okta` metadata namespace, with registry validation enforcing at most one entry per OIN application name and the mapping returned as public catalog data.

+2 more

└──▷ **BREAKING ON UPGRADE**

! The preset selection editor and its Admin API are removed; playbook assignment via `assign_organization_onboarding_playbook` replaces them.

WAS THIS USEFUL?  

DEPLOY

◆ Agentic Infrastructure

Together AI

Feedback ⓘ snapshot-20261002 NOTES ↗ RANK

Run, train, and serve open-source AI models on Together AI.

> Together AI's biggest addition this window is Together Link, a set of terminal wrappers that route popular coding agents through Together-hosted models with 1M-context windows and built-in spend tracking, alongside a new `together-sandbox` SDK/CLI for isolated code execution, a dedicated container inference service with a Queue API, Slurm-based HPC batch GPU scheduling, and extended fine-tuning context for Qwen 27B models.

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

01 together-sandbox SDK and CLI for isolated code execution NEW

82

New `together-sandbox` SDK (Python on PyPI, TypeScript on npm) and standalone CLI run commands and code in isolated Docker-image-snapshot environments, available to allowlisted organizations; environments can be defined via a Docker image or Dockerfile using Together's remote image builder with no local Docker required, and clients authenticate with a Together API key instead of a CodeSandbox API token. Sandboxes are now ephemeral by default — termination is permanent, replacing the legacy hibernate/resume model — so state must be preserved by explicitly snapshotting the filesystem on termination and starting a new sandbox from that snapshot.

Run a command inside an isolated Together sandbox environment using the CLI.

```
$ tcodex exec 'python run_tests.py'
```

— Names SDK, auth method and config, but no direct CLI syntax shown snapshot-20261002

02 Extended fine-tuning context for Qwen 27B BREAKING

75

Fine-tuning context for Qwen 27B models is extended to 131,072 tokens for SFT (up from 32,768) and 65,536 tokens for DPO (up from 16,384), for both LoRA and full fine-tuning. This comes with reduced maximum batch sizes: LoRA fine-tuning drops to a max batch size of 2 (from 16), and DPO full fine-tuning drops to a max batch size of 8 (from 16).

— Exact token and batch-size numbers given, but no command to apply them product docs

THINNER COVERAGE BELOW

03 Dedicated container inference service with Queue API NEW

50

New dedicated container inference service for deploying custom containers on Together's managed GPU infrastructure, with automatic scaling, job queues, and built-in observability. A new Queue API supports submitting, monitoring, and managing asynchronous jobs against these dedicated container inference deployments.

— Names the Queue API and infra but gives no endpoints or commands product docs

04 Batch API file retention window BREAKING

50

Batch API input, output, and error files are now retained for 7 days, enabling reuse of uploaded input files across batch jobs within that window; after 7 days these files are deleted and no longer accessible.

— States the retention window but no API call or config shown product docs

05 Slurm-based HPC batch GPU scheduling NEW

35

Supports HPC-style batch workloads on GPU clusters with Slurm job scheduling, partitions, and job arrays.

— Bare capability description with no commands or config shown snapshot-20261002

└──▷ **BREAKING ON UPGRADE**

! Qwen 27B LoRA fine-tuning jobs now run at a maximum batch size of 2 (down from 16); DPO full fine-tuning maximum batch size drops from 16 to 8.

! In `together-sandbox`, sandboxes are ephemeral by default and termination is permanent — the legacy hibernate and resume model is removed; state must be preserved via explicit filesystem snapshots.

! Batch API files (input, output, error) are deleted after 7 days and are no longer accessible after that window.

> snapshot-20261002 NOTES ↗ LATEST STABLE
>
> Together AI adds HPC-style batch GPU workloads with Slurm scheduling, partitions, and job arrays.
>
> Supports HPC-style batch workloads on GPU clusters with Slurm job scheduling, partitions, and job arrays.
>
> Adds isolated code sandbox runtime environments built from Docker-image snapshots.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.111 NOTES ↗ RANK

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

> HyperFrames pushed six rapid Studio releases focused on audio editing and timeline sync: a new audio/video linking and sync-repair system, an expanded suite of audio mixing tools (gain control, loudness normalization, waveform visualization), new clip presets, a persistent light/dark theme, and deeper host-embedding support for agent tools and undo handling.

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

01 **Audio editing and mixing tools in Studio** NEW 85

Adds audio gain control via pressing `G` on a selected clip. Adds a waveform strip, clip light, peak marks, and video beat source. Adds loudness normalization and duck-under-voice with a limiter report. Adds audible video playback in preview via web audio, plus audio groups, meters, and carve in Core. Adds fade handle visualization on the waveform so fade regions are visible while editing.

– Multiple audio tools named with exact keys and mechanisms v0.8.108 v0.8.110 v0.8.106

02 **Audio/video sync linking and repair in Studio timeline** NEW 75

Adds a `data-link` attribute to link a video clip and its audio track together, enabling linked selection toggle and grouped clip operations. Adds sync-origin tracking and out-of-sync repair with a grouped clip menu in the timeline (part 7/8 of the sync feature series), and completes the audio/video separation work (8/8) with additive selection and partner clamp behaviour.

– Names the data-link attribute and sync mechanisms but no UI path given v0.8.109 v0.8.110

THINNER COVERAGE BELOW

03 **Persistent light/dark theme toggle in Studio** NEW 55

Adds a light and dark theme toggle in the Studio header that persists the user's choice across sessions, and extends the Studio code editor to follow the active light or dark theme.

– Describes persistence and editor scope but no config key v0.8.106

04 **Preset library in Studio clip menu** NEW 50

Adds speed presets, voice presets, look presets, crop presets, and freeze frame to the Studio clip menu.

— Names presets but no behavior or value detail v0.8.108

05 **Host embedding support for undo and agent tools in Studio**

NEW

50

Enables host applications embedding Studio to repaint a still-saving move immediately when the user triggers their own undo key, and enables those embedding hosts to run Studio's agent tools on their own edit session.

— Describes host capabilities but no API surface named v0.8.111 v0.8.107

06 **Clip speed display in Studio timeline** NEW

45

Displays clip playback speed directly in clip names in the Studio timeline.

— Simple display feature with no further mechanism given v0.8.110

v0.8.111 NOTES ↗ LATEST STABLE

HyperFrames v0.8.111 adds instant undo repaint for host-embedded Studio apps during active saves.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.111 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.111
```

Enables host applications embedding Studio to repaint a still-saving move immediately when the user triggers their own undo key.

v0.8.110 NOTES ↗ STABLE

HyperFrames v0.8.110 adds audio gain adjustment with G key and clip speed display in clip names to Studio.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.110 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.110
```

Adds audio gain control in Studio: press `G` on a selected clip to set its gain.

Displays clip playback speed directly in clip names in the Studio timeline.

Adds additive selection and partner clamp behaviour in Studio (completing the audio/video separation work, 8/8).

v0.8.109 NOTES ↗ STABLE

HyperFrames v0.8.109 adds sync-origin detection, out-of-sync repair, linked video/audio selection, and grouped clip menus in Studio.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.109 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.109
```

Adds `data-link` attribute to link a video clip and its audio track together in Studio, enabling linked selection toggle and grouped clip operations.

Adds sync-origin tracking, out-of-sync repair, and grouped clip menu to Studio's timeline (part 7/8 of the sync feature series).

v0.8.108 NOTES ↗ STABLE

HyperFrames v0.8.108 adds waveform strips, loudness normalization with voice ducking, clip menu presets, and audible video preview to Studio.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.108 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.108
```

Adds a waveform strip, clip light, peak marks, and video beat source to Studio.

Adds normalize loudness and duck-under-voice with a limiter report to Studio.

Adds speed presets, voice presets, look presets, crop presets, and freeze frame in the Studio clip menu.

Adds audible video playback in preview via web audio, plus audio groups, meters, and carve in Core.

v0.8.107 NOTES ↗ STABLE

HyperFrames v0.8.107 lets embedded Studio hosts run agent tools on their own edit session and keeps video sound on its clip by default.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.107 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.107
```

v0.8.106 NOTES ↗ STABLE

HyperFrames v0.8.106 adds a persistent light/dark theme toggle in Studio, waveform fade visualization, and write-receipt versioning on the Studio Server.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.106 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.106
```

Adds a light and dark theme toggle in the Studio header that persists the user's choice across sessions.

Extends the Studio code editor to follow the active light or dark theme.

Adds fade handle visualization on the waveform so fade regions are visible while editing.

WAS THIS USEFUL? 👍 👎

Anyscale Ray

Feedback ⓘ ray-2.59.0 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 shipped cluster-wide token authentication that is now enabled by default and required across all nodes as of 2.61, alongside a disk-backed shuffle_v2 backend, GA Ray LLM APIs on vLLM 0.27.0, and a round of Ray Serve and Ray Data additions covering backpressure handling, declarative tracing, scale-to-zero, and new ORC/Lance/video readers.

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

01 Token authentication for Ray clusters

BREAKING — 98

Ray 2.52+ adds `RAY_AUTH_MODE=token` to require a shared secret token for all external Ray APIs and internal cluster communications, resolved via a priority chain of `RAY_AUTH_TOKEN`, `RAY_AUTH_TOKEN_PATH`, and the default `~/.ray/auth_token` file; `ray get-auth-token [--generate]` retrieves or creates a cluster token, and `RAY_AUTH_MODE=disabled` explicitly opts out. Cluster YAML can distribute tokens to remote nodes via `file_mounts` and `initialization_commands` for `ray up`, and `ray dashboard cluster.yaml` sets up secure SSH port forwarding while the dashboard itself prompts for a token and stores it as a cookie for up to 30 days. Ray 2.59 made ray.init() enable token auth by default on local clusters (auto-generating a token at `~/.ray/auth_token`, with `ray start --head` warning if run unauthenticated instead of starting silently), and Ray 2.61 extended default token authentication to all clusters including remote and multi-node deployments, requiring the same token on every node and connecting client or authentication fails.

Generate a token once on your local machine, then start a local Ray cluster with authentication enforced.

```
$ ray get-auth-token --generate
export RAY_AUTH_MODE=token
ray start --head
```

Distribute a locally generated token to all nodes of a self-managed remote cluster before starting Ray with auth enabled.

```
$ ray get-auth-token --generate
scp ~/.ray/auth_token user@node1:~/.ray/auth_token
scp ~/.ray/auth_token user@node2:~/.ray/auth_token
ssh user@node1 "RAY_AUTH_MODE=token ray start --head"
ssh user@node2 "RAY_AUTH_MODE=token ray start --
address=node1:6379"
```

Configure a `ray up` cluster YAML to mount the local token and set `RAY_AUTH_MODE` on every VM for remote authenticated clusters.

```yaml
file_mounts:
  "/home/ubuntu/.ray/auth_token": "~/.ray/auth_token"
initialization_commands:
  - echo "export RAY_AUTH_MODE=token" >> ~/.bashrc
```

Opt out of the new default local-cluster token authentication in environments where it is unwanted (e.g. a fully isolated CI sandbox).

```
$ RAY_AUTH_MODE=disabled python my_ray_job.py
```

— Full auth flow with exact flags, files and runnable commands `ray-2.59.0`

02 **Declarative tracing config for Ray Serve** NEW 81

Adds a `TracingConfig` model (`enabled` , `exporter_import_path` , `sampling_ratio`) that can be passed to serve.start(tracing_config=...) or set in a Serve config file, replacing environment-variable-only tracing setup.

Enable distributed tracing declaratively in a Serve config file instead of relying on environment variables.

```python
from ray import serve
from ray.serve.config import TracingConfig

serve.start(
    tracing_config=TracingConfig(
        enabled=True,

exporter_import_path='opentelemetry.exporter.otlp.proto.grpc.trac
e_exporter:OTLPSpanExporter',
        sampling_ratio=0.1,
    )
)
```

— Includes a runnable code example setting up tracing `ray-2.59.0`

---

03 **Disk-backed shuffle_v2 backend in Ray Data** BREAKING

74

Adds `shuffle_v2`, an external disk-backed shuffle backend with planner integration and Join & Aggregation support, replacing the old `hash_shuffle_v2` strategy name. The compression setting is now controlled via the renamed `shuffle_compression` config key (previously `hash_shuffle_compression`); code or config referencing the old names must be updated.

— Names the config keys but no usage example given `ray-2.59.0`

---

04 **New readers and pipeline controls in Ray Data** NEW

73

Adds native ORC file reading via read_orc(), support for reading multiple paths in read_lance(), and `fps` and `resize` parameters on read_videos() for controlling frame rate and resolution at read time. Also adds `max_concurrent_calls_per_actor` to `ActorPoolStrategy` for per-actor concurrency control, and auto-detects `AzureFileSystem` for Blob HTTPS URLs in download().

— Enumerates every named API addition, no code samples `ray-2.59.0`

---

05 **BackpressureConfig for Ray Serve request queues** NEW

71

Adds a per-deployment `BackpressureConfig` setting in Ray Serve that returns HTTP `429` (with an optional Retry-After header) instead of `503` when `max_queued_requests` is exceeded.

— Clear mechanism and named config but no code sample `ray-2.59.0`

---

06 **Ray Serve deployment API removals and validation** BREAKING

68

Removes `deploy_mode` / `ServeDeployMode`, `use_new_handle_api` on `DeploymentHandle.options`, and the `RAY_AGENT_ADDRESS` deprecation path from Ray Serve. A `#` character in a Ray Serve deployment name now raises a hard `ValueError` since it is used as the replica ID delimiter.

— Lists exact removed APIs but no migration example `ray-2.59.0`

07 **Scale-to-zero and request router extension in Ray Serve** NEW

60

Gang-scheduled deployments can set `min_replicas=0` for scale-to-zero support, and `ray.serve.request_router.__all__` is exposed as a stable public extension API for custom request routers.

— Names both surfaces but gives no usage walkthrough `ray-2.59.0`

THINNER COVERAGE BELOW

08 **Ray LLM APIs reach GA with vLLM 0.27.0** IMPROVED

54

Ray Data LLM and Ray Serve LLM graduate to GA (stable), with the vLLM dependency upgraded to 0.27.0. `HttpRequestUDF` in Ray Serve LLM now supports `multipart/form-data` file uploads.

— Named milestone and version but no usage detail `ray-2.59.0`

└──▷ **BREAKING ON UPGRADE**

- ! Ray 2.59: ray.init() without an `address` enables token authentication by default and generates a token at `~/.ray/auth_token`; set `RAY_AUTH_MODE=disabled` to opt out.
- ! Ray 2.59: `ray start --head` warns and starts unauthenticated if no token is present, rather than starting silently without auth.
- ! Ray 2.61: All clusters (including remote and multi-node) require the same token on every node and every connecting client; clusters without a distributed token will fail to authenticate.
- ! The shuffle strategy name `hash_shuffle_v2` is renamed to `shuffle_v2`; any code or config referencing `hash_shuffle_v2` must be updated.
- ! The compression config key `hash_shuffle_compression` is renamed to `shuffle_compression`; existing config using the old name will break.
- ! `deploy_mode` / `ServeDeployMode`, `use_new_handle_api` on `DeploymentHandle.options`, and the `RAY_AGENT_ADDRESS` deprecation path are removed from Ray Serve.
- ! A `#` character in a Ray Serve deployment name now raises a hard `ValueError` (it is the replica ID delimiter).
- ! ray.init() on a local cluster now enables token authentication by default and writes a token to `~/.ray/auth_token`; set `RAY_AUTH_MODE=disabled` to opt out.

ray-2.59.0 NOTES ↗ LATEST STABLE

Ray 2.59 graduates LLM APIs to GA, ships disk-backed shuffle_v2, token auth by default, and new ORC/Lance/video data sources.

└──▷ GET THIS VERSION

```
$ git clone --branch ray-2.59.0 https://github.com/ray-
project/ray.git
# already have the repo? check out this version:
$ git checkout ray-2.59.0
```

└──▷ TRY IT

Opt out of the new default local-cluster token authentication in environments where it is unwanted (e.g. a fully isolated CI sandbox).

```
$ RAY_AUTH_MODE=disabled python my_ray_job.py
```

Enable distributed tracing declaratively in a Serve config file instead of relying on environment variables.

python

```
from ray import serve
from ray.serve.config import TracingConfig

serve.start(
    tracing_config=TracingConfig(
        enabled=True,

exporter_import_path='opentelemetry.exporter.otlp.proto.grpc.trace_e
xporter:OTLPSpanExporter',
        sampling_ratio=0.1,
    )
)
```

Enables `RAY_AUTH_MODE=disabled` to opt out of the new default token authentication for local clusters; ray.init() now generates a token at `~/.ray/auth_token` automatically.

Adds `shuffle_v2` external disk-backed shuffle backend with planner integration and Join & Aggregation support; controlled via the renamed `shuffle_compression` config key (was `hash_shuffle_compression`).

Adds `BackpressureConfig` per-deployment setting in Ray Serve to return HTTP `429` (with optional `Retry-After` header) instead of `503` when `max_queued_requests` is exceeded.

Adds `TracingConfig` model (`enabled`, `exporter_import_path`, `sampling_ratio`) passable to serve.start(tracing_config=...) or a Serve config file, replacing environment-variable-only tracing setup.

Adds `max_concurrent_calls_per_actor` parameter to `ActorPoolStrategy` for controlling per-actor concurrency in Ray Data pipelines.

+8 more

└─▷ **BREAKING ON UPGRADE**

- ! The shuffle strategy name `hash_shuffle_v2` is renamed to `shuffle_v2`; any code or config referencing `hash_shuffle_v2` must be updated.
- ! The compression config key `hash_shuffle_compression` is renamed to `shuffle_compression`; existing config using the old name will break.
- ! `deploy_mode` / `ServeDeployMode`, `use_new_handle_api` on `DeploymentHandle.options`, and the `RAY_AGENT_ADDRESS` deprecation path are removed from Ray Serve.
- ! A `#` character in a Ray Serve deployment name now raises a hard `ValueError` (it is the replica ID delimiter).
- ! ray.init() on a local cluster now enables token authentication by default and writes a token to `~/.ray/auth_token`; set `RAY_AUTH_MODE=disabled` to opt out.

WAS THIS USEFUL? 👍 👎

Eigen Labs Darkbloom

Feedback ⓘ v0.9.15 NOTES ↗

CODE ↗ RANK

Private Inference Network on Idle Macs

> Darkbloom v0.9.15 adds shadow-mode Autopilot enrollment, provider availability scheduling, automatic idle-engine calibration, a payout rail migration, and a new operator audit tool for withdrawals.

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

01 payout-audit operator CLI for withdrawal auditing NEW

90

A new `payout-audit` operator tool supports bounded withdrawal history auditing and verified refund application via `--apply-refund`, `--verified-stripe-request`, `--expected-amount-micro-usd`, `--limit`, and `--since` flags.

Audit the last 90 days of provider withdrawal history, capped at 100 records per rail, before applying any refund.

```
$ payout-audit --limit 100 --since 2025-01-01T00:00:00Z
```

Apply a verified refund for a rejected Connect withdrawal after an operator has confirmed the Stripe request ID and exact amount.

```
$ payout-audit --apply-refund a96a14d8-360a-4dd9-8c41-1dde880bab73 --expected-amount-micro-usd 1000000 --verified-stripe-request req_verified
```

— Runnable CLI examples with every flag named v0.9.15

THINNER COVERAGE BELOW

02 Automatic MiMo engine calibration NEW

55

Idle MiMo engines are automatically calibrated after model loading, refreshing stale phase evidence with uncached built-in prompts and publishing measurements through existing heartbeats.

— Mechanism described but no command to invoke v0.9.15

03 Shadow-mode Autopilot enrollment NEW

50

`StartAutopilotController` enrolls downloaded network models in shadow-mode Autopilot, observing routing decisions without affecting live traffic.

— Named function but no usage example given v0.9.15

04 **Global Payouts migration for bank payouts** IMPROVED

30

Bank payouts are migrated to Global Payouts while retaining Stripe settlement, broadening payout reach for providers.

— States change but no mechanism or reader action v0.9.15

05 **Interactive availability scheduling for providers** NEW

25

Providers can use interactive availability scheduling to signal when their nodes are available for inference requests.

— Thin description with no named surface or example v0.9.15

v0.9.15 NOTES ↗ CODE ↗ LATEST STABLE

Darkbloom v0.9.15 adds interactive availability scheduling, shadow-mode Autopilot enrollment, and automatic MiMo engine calibration for providers.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.9.15 https://github.com/Layr-Labs/d-inference.git
# already have the repo? check out this version:
$ git checkout v0.9.15
```

└──▷ TRY IT

Audit the last 90 days of provider withdrawal history, capped at 100 records per rail, before applying any refund.

```
$ payout-audit --limit 100 --since 2025-01-01T00:00:00Z
```

Apply a verified refund for a rejected Connect withdrawal after an operator has confirmed the Stripe request ID and exact amount.

```
$ payout-audit --apply-refund a96a14d8-360a-4dd9-8c41-1dde880bab73 --expected-amount-micro-usd 1000000 --verified-stripe-request req_verified
```

Adds `StartAutopilotController` to enroll downloaded network models in shadow-mode Autopilot, observing routing decisions without affecting live traffic.

Adds interactive availability scheduling for providers, allowing nodes to signal when they are available for inference requests.

Enables automatic calibration of idle MiMo engines after model loading, refreshing stale phase evidence with uncached built-in prompts and publishing measurements through existing heartbeats.

Migrates bank payouts to Global Payouts with retained Stripe settlement, broadening payout reach for providers.

Adds `payout-audit` operator tool with `--apply-refund`, `--verified-stripe-request`, `--expected-amount-micro-usd`, `--limit`, and `--since` flags for bounded withdrawal history auditing and verified refund application.

WAS THIS USEFUL? 👍 👎

Fireworks AI

Feedback ⓘ

Managed RFT is paused; try RL on the Training API · NOTES ↗ · RANK

Fireworks AI provides hosted inference, model fine-tuning, and deployment APIs for open-weight models.

> Fireworks AI replaced Managed RFT with full-parameter Reinforcement Learning fine-tuning on its Training API (up to 524K context, on-policy distillation), paused managed RFT entirely for 70 models, broadened SFT/DPO model family support, and added DeepSeek V4.1 Flash, GLM 5.3, and GLM 5.3 Flash to its Dedicated and Serverless Training APIs.

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

01 Reinforcement Learning fine-tuning via the Training API NEW

85

Fireworks adds Reinforcement Learning fine-tuning via the Training API, superseding Managed RFT: users write their own rollout and training loop to train against a reward function, with support for full-parameter RL (not just LoRA), up to 524K token context length, on-policy distillation (OPD), and custom objectives.

— Mechanism and limits named, but no exact API call shown

Managed RFT is paused; try RL on the Tr…

02 Managed RFT paused across UI, firectl and REST API BREAKING

75

Managed RFT no longer accepts new jobs via the Fireworks UI, `firectl`, or the REST API; existing jobs remain visible and already-trained models continue serving. 70 models that were tunable only via managed RFT — including `deepseek-r1`, `deepseek-v3`, `kimi-k2-instruct`, `qwen3-235b-a22b`, `llama-v3-70b-instruct`, `gemma-3-27b-it`, and `gpt-oss-120b` — can no longer be fine-tuned on Fireworks and no longer appear on the Models page. Inference on these models is unaffected, and Managed SFT and DPO are unaffected.

— Full scope and model list given, but no migration command

Managed RFT is paused; try RL on the Tr… · New training model: DeepSeek V4.1 Flash · New Serverless Training model: GLM 5.3 … · New Serverless Training model: GLM 5.3 · New training model: GLM 5.3 Flash · New training model: GLM 5.3

03 New models added to Training API NEW

70

Fireworks added new models to its Training API for fine-tuning: `DeepSeek V4.1 Flash` (Dedicated Training API, LoRA, up to 262K context), `GLM 5.3 Flash`

(Serverless Training, 200K context, vision; also added to the Dedicated Training API with LoRA and vision support, up to 262K context), and `GLM 5.3` (Serverless Training, LoRA, up to 262K context; also added to the Dedicated Training API, LoRA, up to 204K context, optimized for complex coding and long-horizon agents).

— Models, platforms and context limits named but no command shown

New training model: GLM 5.3 Flash | New Serverless Training model: GLM 5.3 | New Serverless Training model: GLM 5.3 … | New training model: DeepSeek V4.1 Flash | New training model: GLM 5.3

THINNER COVERAGE BELOW

04 **SFT/DPO model family support extended** IMPROVED 55

Fine-tuning model family support for SFT and DPO now extends to `DeepSeek`, `Qwen`, `Kimi`, `Gemma`, `GLM`, and `Llama`, with eligibility decided per model and per method.

— Names families but not which specific models are eligible product docs

└──▷ BREAKING ON UPGRADE

- ! Managed RFT is deprecated; Reinforcement Learning fine-tuning is now available through the Training API instead.
- ! Managed RFT is paused: new RFT jobs can no longer be submitted via the Fireworks UI, `firectl`, or the REST API.
- ! 70 models (including `deepseek-r1`, `deepseek-v3`, `kimi-k2-instruct`, `qwen3-235b-a22b`, `llama-v3-70b-instruct`, `gemma-3-27b-it`, and others) can no longer be fine-tuned on Fireworks and no longer appear on the Models page.
- ! Managed RFT is paused: the Fireworks UI, `firectl`, and the REST API no longer accept new RFT jobs, and the 70 models that were tunable only via managed RFT (including `deepseek-r1`, `deepseek-v3`, `qwen3-235b-a22b`, `llama-v3-70b-instruct`, and others — see full list in release notes) can no longer be fine-tuned on Fireworks.
- ! Managed RFT is paused: Fireworks UI, `firectl`, and the REST API no longer accept new managed RFT jobs, so any pipeline that submits managed RFT jobs will stop working. The 70 models that were tunable only via managed RFT (including `deepseek-r1`, `qwen3-235b-a22b`, `llama-v3-70b-instruct`, `kimi-k2-instruct`, and others) can no longer be fine-tuned on Fireworks.
- ! Managed RFT is paused: the Fireworks UI, `firectl`, and the REST API no longer accept new RFT jobs.
- ! 70 models that were tunable only through managed RFT — including `deepseek-r1`, `kimi-k2-instruct`, `qwen3-235b-a22b`, `llama-v3-70b-instruct`, `gpt-oss-120b`, and others — can no longer be fine-tuned on Fireworks and no longer appear on the Models page (inference on these models is unaffected).
- ! Managed RFT (reinforcement fine-tuning) is paused: `firectl`, the Fireworks UI, and the REST API no longer accept new RFT jobs. The 70 models that were tunable only via managed RFT — including `deepseek-r1`, `qwen3-235b-a22b`, `llama-v3-70b-instruct`,

`kimi-k2-instruct`, and others — can no longer be fine-tuned on Fireworks and no longer appear on the Models page. Inference on these models is unaffected.

! Managed reinforcement fine-tuning (RFT) is paused — the Fireworks UI, `firectl`, and the REST API no longer accept new RFT jobs, and 70 models (including `deepseek-r1`, `qwen3-235b-a22b`, `llama-v3-70b-instruct`, `kimi-k2-instruct`, and others) can no longer be fine-tuned on Fireworks. Inference on these models is unaffected. Managed SFT and DPO are unaffected.

`Managed RFT is paused; try RL on the Training API` NOTES ↗ LATEST STABLE

Managed RFT is paused; Training API now offers full-parameter RL with up to 524K token context and on-policy distillation.

Adds RL via the Training API where you write your own rollout and training loop, supporting full-parameter RL (not just LoRA), up to 524K token context length, on-policy distillation (OPD), and custom objectives.

Managed Training no longer accepts new RFT jobs from the Fireworks UI, `firectl`, or the REST API; existing jobs remain visible and already-trained models continue serving.

└──▷ **BREAKING ON UPGRADE**

! Managed RFT is paused: new RFT jobs can no longer be submitted via the Fireworks UI, `firectl`, or the REST API.

! 70 models (including `deepseek-r1`, `deepseek-v3`, `kimi-k2-instruct`, `qwen3-235b-a22b`, `llama-v3-70b-instruct`, `gemma-3-27b-it`, and others) can no longer be fine-tuned on Fireworks and no longer appear on the Models page.

WAS THIS USEFUL? 👍 👎

Modal

Feedback ⓘ changelog · Oct 2, 2026 CHANGELOG ↗ RANK

Bring your own code, and run CPU, GPU, and data-intensive compute at scale. The serverless platform for AI and data teams.

> Modal shipped multi-node clusters and sticky sessions for Servers, a new full-VM runtime for Sandboxes enabling Docker-in-Sandbox workloads, and a broad set of CLI/SDK introspection commands for Functions, Servers and Apps, alongside breaking changes removing legacy Sandbox filesystem APIs and changing Sandbox creation and CLI JSON-output behavior.

└──▷ **WHAT SHIPPED · 17 FEATURES** most completely described first ? what's the number?

01 **VM and gVisor runtimes for Sandboxes** NEW 91

Adds a `runtime` parameter to modal.Sandbox.create() accepting `'vm'` or `'gvisor'`, letting users choose between the default gVisor isolated containers and a full Linux VM. The VM runtime supports running Docker inside a Sandbox (via `dockerd` as the entrypoint), mounting FUSE filesystems, and managing resources with nested cgroups, with nested virtualization available on Team and Enterprise plans. Also adds a `--runtime` option to the `modal shell` CLI command for selecting the Sandbox runtime, and Sandboxes now use a next-generation backend by default, enabling substantially higher creation rates and concurrency limits for CPU Sandboxes with no code changes required.

Spin up a Sandbox with a full VM runtime to run Docker commands inside it.

```
$ modal sandbox create --runtime vm ubuntu:24.04 -- docker run hello-world
```

Run Docker containers inside a Modal Sandbox for CI pipelines that need a full Docker stack.

```python
image = (
    modal.Image.from_registry('ubuntu:24.04')
    .env({'DEBIAN_FRONTEND': 'noninteractive'})
    .apt_install('docker.io')
)

sb = modal.Sandbox.create(
    'dockerd',
    image=image,
    runtime='vm',
    readiness_probe=modal.Probe.with_exec('docker', 'info',
interval_ms=500),
)

p = sb.exec('docker', 'run', '--rm', 'alpine', 'echo', 'hello
from Docker')
print(p.stdout.read())
```

Switch a Sandbox to the gVisor runtime for stronger isolation on untrusted workloads.

```python
sb = modal.Sandbox.create(app=sb_app, runtime='gvisor')
p = sb.exec('uname', '-srn')
print(p.stdout.read())
```

Launch a Sandbox running Docker-in-Docker by selecting the full VM runtime instead of gVisor.

```python
import modal
sb = modal.Sandbox.create(image=modal.Image.debian_slim(),
runtime="vm")
```

— Mechanism, runtime options, and runnable code/CLI examples given

changelog-20261002-325f79ab 1.6.0 (2026-09-28)

02 **Introspection CLI and SDK for Functions, Servers, Apps** NEW

85

Adds `modal function` and `modal server` CLI surfaces with `info`, `stats`, `logs`, and `calls` / `requests` subcommands for inspecting Function and Server entities, plus corresponding SDK methods on `modal.Function` and `modal.Server`. Adds `modal app info` CLI and modal.App.info() SDK method for inspecting Apps,

`modal function variants` CLI to list parameterizations and dynamic configurations of a Function, modal.Environment.apps.list() for programmatic App listing, and a `--limit` option to `modal app list` CLI, which now also orders Apps by recency.

Issue a short-lived token and check live request stats for a deployed Server from the CLI.

```
$ modal token new --expires-in 3600
modal server stats my-app::MyServer
```

— Every subcommand and SDK method named with a runnable example `1.6.0 (2026-09-28)`

03 **Sticky sessions for Servers** NEW 76

Adds @modal.sessioned() decorator plus Server.sessions.start() and Server.sessions.terminate() methods for sticky sessions with strong affinity, enabling WebSocket workloads that must reconnect to the same container via a `Modal-Authorization` header token.

— Mechanism and header token named, no code example `1.6.0 (2026-09-28)`

04 **Legacy Sandbox filesystem APIs removed** BREAKING 73

The legacy Sandbox filesystem APIs Sandbox.open(), Sandbox.ls(), Sandbox.mkdir(), Sandbox.rm(), and Sandbox.watch(), the associated `modal.file_io.FileIO` type, and `modal.exception.FilesystemExecutionError` have been removed; use the `Sandbox.filesystem` APIs instead.

— All removed APIs named with migration path `changelog-20261002-325f79ab` `1.6.0 (2026-09-28)`

05 **Token TTL and named proxy tokens** NEW 70

Adds `--expires-in` option to `modal token new` CLI to set a TTL for new tokens, and named proxy tokens via `modal workspace proxy-tokens` CLI and `modal.Workspace.proxy_tokens` SDK interface.

Issue a short-lived token and check live request stats for a deployed Server from the CLI.

```
$ modal token new --expires-in 3600
modal server stats my-app::MyServer
```

Create a short-lived API token that expires in 24 hours, useful for temporary CI/CD access.

```
$ modal token new --expires-in 24h
```

— Flags named with runnable CLI examples changelog-20261002-325f79ab 1.6.0 (2026-09-28)

06 **Concurrency limit for Servers** NEW 70

Adds `max_concurrency=` parameter to @app.server() decorator to set a strict concurrency limit, rejecting excess requests with HTTP 503.

— Named parameter and exact rejection behavior, no code example 1.6.0 (2026-09-28)

07 **JSON schema changes for `modal volume ls` and `modal queue list`**

68

BREAKING

`modal volume ls` JSON output now shows `created_modified` as an ISO 8601 timestamp and `size` as an integer number of bytes (previously different types). `modal queue list` JSON output now shows `partitions` and `total_size` as integers and adds a boolean `total_size_truncated` field.

— Every changed field and type named changelog-20261002-325f79ab 1.6.0 (2026-09-28)

08 **Multi-node clusters for Functions and Servers** NEW

65

Adds @modal.clustered() decorator and `modal.Cluster` object to run Functions and Servers on ephemeral multi-node clusters for large-scale training or inference, exposing each container's rank and private IPs.

— Mechanism and named decorator given, no runnable example 1.6.0 (2026-09-28)

09 **Readiness probes for Sandbox exec calls** NEW 60

Adds modal.Probe.with_exec() as a `readiness_probe` option on modal.Sandbox.create(), gating exec() calls until a service inside the Sandbox is ready.

Run Docker containers inside a Modal Sandbox for CI pipelines that need a full Docker stack.

```python
image = (
    modal.Image.from_registry('ubuntu:24.04')
    .env({'DEBIAN_FRONTEND': 'noninteractive'})
    .apt_install('docker.io')
)

sb = modal.Sandbox.create(
    'dockerd',
    image=image,
    runtime='vm',
    readiness_probe=modal.Probe.with_exec('docker', 'info',
interval_ms=500),
)

p = sb.exec('docker', 'run', '--rm', 'alpine', 'echo', 'hello
from Docker')
print(p.stdout.read())
```

— Named method and parameter shown in a code example `product docs`

10 **Sandbox creation now blocks until scheduled** BREAKING

60

modal.Sandbox.create() now blocks until the Sandbox is scheduled instead of returning immediately; if scheduling fails (e.g. due to unsatisfiable resource or placement requests), it raises `ResourceExhaustedError`.

— Exact before/after behavior and exception named `changelog-20261002-325f79ab` `1.6.0 (2026-09-28)`

11 **`@app.cls()` no longer supports custom `__init__`** BREAKING

60

Classes decorated with @app.cls() can no longer define a custom `__init__` constructor; use modal.parameter() for parameters and @modal.enter() for initialization logic.

— Exact migration path named `1.6.0 (2026-09-28)`

THINNER COVERAGE BELOW

12 **Higher call graph node limit** IMPROVED

45

Raises FunctionCall.get_call_graph() node limit from 100 to 5,000.

— Specific numeric change, no example `1.6.0 (2026-09-28)`

13 **`modal bootstrap` CLI removed** BREAKING 45

The experimental `modal bootstrap` CLI is removed; use `modal endpoint create` instead.

– Replacement command named, no further detail `1.6.0 (2026-09-28)`

14 **Sandbox log retrieval API** NEW 40

Adds `modal.Sandbox.logs` API supporting retrieval of logs by date/time range and tail of the most recent logs.

– Named API but no usage example given `changelog-20261002-325f79ab`

15 **Global --profile CLI option** NEW 40

Adds a global `--profile` option to the Modal CLI for ad hoc profile selection.

– Flag named but no usage shown `changelog-20261002-325f79ab`

16 **Default role for Restricted Environments** NEW 35

Adds support for configuring the default role when creating a new Restricted Environment via the CLI or SDK.

– No flag or field names given, generic prose `changelog-20261002-325f79ab`

17 **Undocumented properties and methods removed** BREAKING 9

Several undocumented properties and methods with no intended user-facing functionality have been removed.

– No names given, vague cleanup note `changelog-20261002-325f79ab`

└──▷ BREAKING ON UPGRADE

! The legacy Sandbox filesystem APIs Sandbox.open(), Sandbox.ls(), Sandbox.mkdir(), Sandbox.rm(), and Sandbox.watch(), the associated file type, and `modal.exception.FilesystemExecutionError` have been removed; use the `Sandbox.filesystem` APIs instead.

! Several undocumented properties and methods with no intended user-facing functionality have been removed.

! `modal volume ls` JSON output now shows `created_modified` as an ISO 8601 timestamp and `size` as an integer number of bytes (field types changed).

! `modal queue list` JSON output now shows `partitions` and `total_size` as integers and adds a boolean `total_size_truncated` field (field types and schema changed).

! If a Sandbox cannot be scheduled (e.g. due to unsatisfiable resource or placement requests), creation now times out and raises a `ResourceExhaustedError` instead of returning immediately.

! Classes decorated with @app.cls() can no longer define a custom `__init__` constructor; use modal.parameter() for parameters and @modal.enter() for initialization logic.

! Sandbox.open(), Sandbox.ls(), Sandbox.mkdir(), Sandbox.rm(), Sandbox.watch(), the `modal.file_io.FileIO` type, and the `modal.exception.FilesystemExecutionError` exception are removed; use the `Sandbox.filesystem` APIs instead.

! The experimental `modal bootstrap` CLI is removed; use `modal endpoint create` instead.

! `modal volume ls` JSON output now shows `created_modified` as an ISO 8601 timestamp and `size` as an integer number of bytes (previously different types).

! `modal queue list` JSON output now shows `partitions` and `total_size` as integers and adds a boolean `total_size_truncated` field.

! modal.Sandbox.create() now blocks until the Sandbox is scheduled instead of returning immediately; if scheduling fails, it raises `ResourceExhaustedError`.

changelog · Oct 2, 2026 CHANGELOG ↗ LATEST STABLE

Modal: Adds `runtime="vm"` option to Sandboxes to run inside a full virtual machine (supports workflows like running Docker); use `runtime="gvisor"` to keep the current default.

└──▷ **TRY IT**

Spin up a Sandbox with a full VM runtime to run Docker commands inside it.

```
$ modal sandbox create --runtime vm ubuntu:24.04 -- docker run
hello-world
```

Issue a short-lived token and check live request stats for a deployed Server from the CLI.

```
$ modal token new --expires-in 3600
modal server stats my-app::MyServer
```

Adds `modal.Sandbox.logs` API supporting retrieval by date/time range and tail of most recent logs.

Adds support for configuring the default role when creating a new Restricted Environment via the CLI or SDK.

Adds a global `--profile` option to the Modal CLI for ad hoc profile selection.

└──▷ **BREAKING ON UPGRADE**

! The legacy Sandbox filesystem APIs Sandbox.open(), Sandbox.ls(), Sandbox.mkdir(), Sandbox.rm(), and Sandbox.watch(), the associated file type, and `modal.exception.FilesystemExecutionError` have been removed; use the `Sandbox.filesystem` APIs instead.

! Several undocumented properties and methods with no intended user-facing functionality have been removed.

! `modal volume ls` JSON output now shows `created_modified` as an ISO 8601 timestamp and `size` as an integer number of bytes (field types changed).

! `modal queue list` JSON output now shows `partitions` and `total_size` as integers and adds a boolean `total_size_truncated` field (field types and schema changed).

! If a Sandbox cannot be scheduled (e.g. due to unsatisfiable resource or placement requests), creation now times out and raises a `ResourceExhaustedError` instead of returning immediately.

1.6.0 (2026-09-28) NOTES ↗ STABLE

Modal 1.6.0 adds multi-node clusters, VM sandboxes, sticky sessions, and a new CLI/SDK observability surface.

└──▷ **USE IT**

Launch a Sandbox running Docker-in-Docker by selecting the full VM runtime instead of gVisor.

```python
import modal
sb = modal.Sandbox.create(image=modal.Image.debian_slim(),
runtime="vm")
```

Create a short-lived API token that expires in 24 hours, useful for temporary CI/CD access.

```
$ modal token new --expires-in 24h
```

Adds @modal.clustered() decorator and `modal.Cluster` object to run Functions and Servers on ephemeral multi-node clusters for large-scale training or inference, exposing each container's rank and private IPs.

Adds @modal.sessioned() decorator plus Server.sessions.start() and Server.sessions.terminate() methods for sticky sessions with strong affinity, enabling WebSocket workloads that must reconnect to the same container via a `Modal-Authorization` header token.

Adds `runtime="vm"` and `runtime="gvisor"` parameters to modal.Sandbox.create() to run Sandboxes in a full VM (better Docker support) or force the gVisor runtime.

Adds `--runtime` option to the `modal shell` command for selecting the Sandbox runtime from the CLI.

Adds `modal function` and `modal server` CLI surfaces with `info`, `stats`, `logs`, and `calls` / `requests` subcommands for inspecting Function and Server entities.

+10 more

└──▷ **BREAKING ON UPGRADE**

- ! Classes decorated with @app.cls() can no longer define a custom `__init__` constructor; use modal.parameter() for parameters and @modal.enter() for initialization logic.
- ! Sandbox.open(), Sandbox.ls(), Sandbox.mkdir(), Sandbox.rm(), Sandbox.watch(), the `modal.file_io.FileIO` type, and the `modal.exception.FilesystemExecutionError` exception are removed; use the `Sandbox.filesystem` APIs instead.
- ! The experimental `modal bootstrap` CLI is removed; use `modal endpoint create` instead.
- ! `modal volume ls` JSON output now shows `created_modified` as an ISO 8601 timestamp and `size` as an integer number of bytes (previously different types).
- ! `modal queue list` JSON output now shows `partitions` and `total_size` as integers and adds a boolean `total_size_truncated` field.
- ! modal.Sandbox.create() now blocks until the Sandbox is scheduled instead of returning immediately; if scheduling fails, it raises `ResourceExhaustedError`.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

SGLang

Feedback ⓘ v0.5.21 NOTES ↗ CODE ↗ RANK

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

> SGLang shipped prefill context parallelism for long-prompt serving, attention data parallelism (`--attn-dp-size`) that deprecates the old `--dp-size --enable-dp-attention` combo, and a v0.5.21 release adding a built-in diffusion engine, new `/v1/decisions` and `/v1/score` classifier APIs, live PD role switching, a Rust-backed prefix cache, and support for ten new models.

└──▷ WHAT SHIPPED · 14 FEATURES most completely described first ? what's the number?

01 **Prefill context parallelism across GPUs** NEW 98

Adds `--enable-prefill-cp` to distribute long-prompt prefill tokens across a GPU attention CP group, with `--cp-strategy` (`zigzag` or `interleave`) controlling how tokens are split and `--attn-cp-size` setting the CP group size (TP world partitioned as attention DP size × attention CP size × attention TP size). Adds `--enable-dsa-cache-layer-split` for DSA deployments (requires `--disaggregation-mode prefill`, `--enable-prefill-cp --cp-strategy interleave`, `PP=1`; supports `mooncake` / `mooncake_tcp` transfer backends) to shard GPU KV/indexer cache layers across CP ranks and cut per-rank memory use. Introduces a `ContextParallelStrategy` interface separating token-layout policy from model execution, with per-forward state in `ForwardBatch.attn_cp_metadata`; composes with TP, DP, EP, PP, and speculative decoding, plus an experimental CUDA graphs path (zigzag, `trtllm_mha`, `PP=1`, CP size = TP size). Interleave suits DSA models like GLM-5 and DeepSeek V3.2/V4, zigzag suits GQA models with FlashAttention or `trtllm_mha`.

Serve a Qwen3 MoE model across 4 Hopper GPUs with attention `CP=2` and `TP=2` using the zigzag strategy to parallelize long-prompt prefill.

```
$ python -m sglang.launch_server \
  --model-path Qwen/Qwen3-30B-A3B-FP8 \
  --tp 4 \
  --ep 4 \
  --attn-cp-size 2 \
  --enable-prefill-cp \
  --cp-strategy zigzag \
  --attention-backend fa3
```

Serve a DSA-based GLM-5.2 model across 8 H200 GPUs with attention `CP=8` and the interleave strategy required for sparse-attention (DSA) models.

```
$ python -m sglang.launch_server \
  --model-path zai-org/GLM-5.2-FP8 \
  --tp 8 \
  --attn-cp-size 8 \
  --enable-prefill-cp \
  --cp-strategy interleave \
  --attention-backend dsa \
  --trust-remote-code \
  --mem-fraction-static 0.8
```

Enable layer-wise KV/indexer cache sharding across CP ranks in a PD-disaggregated DSA deployment to reduce per-rank GPU memory pressure.

```
$ python -m sglang.launch_server \
  --model-path zai-org/GLM-5.2-FP8 \
  --tp 8 \
  --attn-cp-size 8 \
  --enable-prefill-cp \
  --cp-strategy interleave \
  --disaggregation-mode prefill \
  --enable-dsa-cache-layer-split \
  --attention-backend dsa \
  --trust-remote-code
```

— Full flag set, strategy options, and runnable launch commands given product docs

02

LiLiCorr reranking and DFlash speculative decoding tuning

IMPROVED

88

Adds `SGLANG_ENABLE_LILICORR_SAMPLING=1` to enable temperature-aware draft sampling for LiLiCorr, sampling each slot from softmax(scores/T) over candidates at the request's temperature instead of committing the argmax, worth ~6% acceptance and ~6% end-to-end throughput on Qwen3-8B at temperature 1. Adds an `AttentionBackend.needs_cpu_seq_lens` property (default `True`; backends `triton`, `trtllm_mha`, `trtllm_mla`, `flashmla`, `dsa` set it `False`) letting backends skip the per-block device-to-host `seq_lens_cpu` sync that dominates throughput in 16-slot DFlash decoding. Adds grouped dynamic convolution composition via `conv_kernel_size` and `conv_group_size` keys in `dflash_config` (both default `1`), with the loader refusing checkpoints whose convolution tensors and built modules disagree. v0.5.21 adds LiLiCorr itself as a candidate-lattice reranker for DFlash drafts.

— Env var, property, and config keys with concrete benchmark numbers v0.5.21

03 Attention data parallelism via --attn-dp-size BREAKING

83

Adds `--attn-dp-size N` (alias `--attention-data-parallel-size`) to enable Attention Data Parallelism, creating N attention data-parallel groups inside the TP group each with its own scheduler while FFN stays tensor parallel; elastic EP scale-up joiners run DPA automatically even when launched with `--attn-dp-size 1`. This replaces the deprecated `--dp-size N --enable-dp-attention` combination, which still resolves to `--attn-dp-size N --dp-size 1` with a warning. The `/server_info` endpoint now reports `attn_dp_size` directly, and its `enable_dp_attention` field is deprecated and always returns `false`.

Launch an SGLang server with 8 attention data-parallel groups inside the TP group to enable DPA for high-throughput inference.

```
$ python -m sglang.launch_server --attn-dp-size 8
```

Query `/server_info` to confirm DPA is active and read the authoritative `attn_dp_size` value instead of the deprecated `enable_dp_attention` field.

```
$ curl http://localhost:30000/server_info | jq '{attn_dp_size:
.attn_dp_size}'
```

Launch a TP-sharded model with 4 attention-DP groups to increase scheduling throughput while keeping FFN computation tensor-parallel.

```
$ python -m sglang.launch_server --model-path <model> --tp-size 8
--attn-dp-size 4
```

— Named flag, alias, deprecation path, and API field migration product docs

04 FFN boundary reduction control and layer boundary API NEW

70

Adds `--boundary-reduction` to select the FFN output reduction strategy (`ar`, `rs`, `rsv`, `rs+rsv`) for decoders built with `declare_ffn` / `make_stages` stage boundaries, with model-aware defaults per backbone, plus a `speculative_boundary_reduction` server-args field (no CLI flag) recording the resolved policy after startup. Adds a layer boundary API for declaring stage inputs and

outputs, binding communication and residual operations, and integrating model computation.

Enable variable-size reduce-scatter on MoE attention-DP paths to reduce communication overhead for eligible MoE models.

```
$ python -m sglang.launch_server --model-path <model> --attn-dp-size 4 --boundary-reduction rsv
```

Use reduce-scatter with fallback to all-reduce for a Qwen3-MoE model that would otherwise default to `rs+rsv`, to test fixed-size paths explicitly.

```
$ python -m sglang.launch_server --model-path Qwen/Qwen3-MoE-A3B --attn-dp-size 4 --boundary-reduction rs
```

— Named flag with values and field, API itself not demonstrated product docs

05 **New model support and deployment cookbooks** NEW 70

Adds support for new models DeepSeek-V4.1 Flash, GigaChat 3.5, IQuest-Q1, MiMo-V2.6/MiMo-V2.6-Pro, Ling-3.0-flash-VL, DiffusionGemma, Qwen-Image 2.1, Anima Base v1.0, Ming-Image 0.1 Design/Design-Layer, and FLUX 3 Action. Adds deployment cookbooks for IQuest-Q1 (320B MoE) on 8× NVIDIA H200 in BF16 with optional MTP speculative decoding; GigaChat 3.5 (432B-A28B hybrid MLA + GDN MoE) on H100/H200 in FP8 with MTP speculative decoding and tool calling; GigaChat 3.5 Reasoning (432B-A28B hybrid MLA + GDN MoE thinking model) on H100/H200 in FP8 with reasoning parsing, MTP speculative decoding, and tool calling; GLM-5.3-Flash on H100, H200, B200, B300, GB200, GB300, and MI355X with MTP and multimodal serving; and FLUX 3 Action (vision-language-action model).

— Extensive named models and hardware, no exact commands given v0.5.21

06 **SGLang Diffusion engine for image and video generation** NEW 68

Adds SGLang Diffusion, a built-in inference engine for image and video generation shipping in the same repository and package, installable via Docker, pip, uv, or from source. Supports a diffusers backend alongside native SGLang pipelines, exposes an OpenAI-compatible server interface, and adds a `--host` flag for starting the HTTP server in diffusion mode. Ships a ComfyUI plugin for using SGLang from ComfyUI in server mode or as a per-step DiT backend, adds support for running MiniMax-H3 with SGLang Diffusion inside ComfyUI, and adds opt-in SRT prompt enhancement to the image and video diffusion APIs.

— Install methods and flag named, deeper mechanism not described v0.5.21

07 **Classifier and scoring APIs** NEW 64

Adds `/v1/decisions`, an API endpoint that turns an LLM or VLM into a low-latency classifier and scorer. Adds `/v1/score`, which scores all candidates in a single request with per-item candidate token IDs, temperature scaling, and optional full-vocabulary token logprobs.

— Endpoint paths and fields named, no example calls given v0.5.21

THINNER COVERAGE BELOW

08 **Speculative decoding expansion across models and kernels** NEW 55

Adds pipeline parallelism × speculative decoding (EAGLE/MTP) compatibility. Adds DSpark speculative decoding for LFM2-VL (1.66x–2.56x speedup at batch 1). Adds DFLASH speculative decoding support for Kimi K3. Adds SM100 NVFP4 GenMHA and speculative decoding for KV-cache, giving a 1.355x faster decode attention kernel at 1M context.

— Named targets and speedup numbers but no invocation shown v0.5.21

09 **Decode latency and throughput improvements** IMPROVED 45

Reduces decode bootstrap latency with request-owned speculative KV, dropping mean TTFT from 19.4 s to 15.1 s on DeepSeek-V4-Pro 1P1D on MI355X at concurrency 256. DeepSeek-V4.1 achieves 22% faster first token on long prompts, and Kimi K3 achieves 20.6% higher prefill throughput in PD serving.

— Concrete benchmark numbers given but no reproduction steps v0.5.21

10 **Prefill-decode disaggregation enhancements** NEW 41

Introduces runtime role switching for PD instances so a node can switch between prefill and decode on the fly without a restart, and adds optional KV transfer checksums for prefill-decode disaggregation.

— Described in prose, no config or flag named v0.5.21

11 **BF16 and batch-invariant inference with DeepEP v2** NEW 35

Adds BF16 and batch-invariant inference support with DeepEP v2, including MXFP8 dispatch and deferred route weighting.

— Named techniques, no flags or benchmark context given v0.5.21

12 **Unified per-token-slot paging for Mamba and KV caches** NEW 30

Supports unified per-token-slot paging that stores Mamba state and full/SWA KV caches as a single page-granular buffer instead of one buffer per layer, requiring stride-aware attention, linear-attention, and Mamba backends.

— No flag or config key named, mechanism only `product docs`

13 **8-node MSCCL++ collective algorithms** NEW 25

Adds 8-node AllReduce/AllGather and MNVLS algorithm support to MSCCL++.

— Named algorithm support only, no usage detail `v0.5.21`

14 **Rust-backed prefix cache by default** IMPROVED 19

Switches the prefix cache to a Rust core by default.

— One-line mention, no mechanism or migration detail `v0.5.21`

└──▷ **BREAKING ON UPGRADE**

! The `enable_dp_attention` field in `/server_info` responses is deprecated and always returns `false`; consumers must read `attn_dp_size` instead.

! The `--dp-size N --enable-dp-attention` flag combination is deprecated; it still resolves to `--attn-dp-size N --dp-size 1` with a deprecation warning, but users should migrate to `--attn-dp-size N`.

! The `--attn-dp-size <dp-size>` flag replaces the `--dp-size <dp-size> --enable-dp-attention` combination; existing launch scripts using the old flags must be updated.

`v0.5.21` NOTES ↗ CODE ↗ LATEST STABLE

SGLang v0.5.21 adds `/v1` /decisions and `/v1` /score APIs, live PD role switching, Rust-backed prefix cache, and 10 new models.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.5.21 https://github.com/sgl-
project/sglang.git
# already have the repo? check out this version:
$ git checkout v0.5.21
```

Adds `/v1/decisions` API endpoint that turns an LLM or VLM into a low-latency classifier and scorer.

Adds `/v1/score` API endpoint that scores all candidates in a single request, with per-item candidate token IDs, temperature scaling, and optional full-vocabulary token logprobs.

Introduces runtime role switching for PD instances so a node can switch between prefill and decode on the fly without a restart.

> Adds optional KV transfer checksums for prefill-decode disaggregation.
>
> Switches the prefix cache to a Rust core by default.
>
> +13 more

WAS THIS USEFUL?  

Mintplex Labs AnythingLLM

Feedback ⓘ v1.17.0

NOTES ↗ RANK

AnythingLLM is a local-first desktop and server application that runs agents, RAG workspaces, and MCP tools against local or hosted models.

> AnythingLLM v1.17.0 expands agent and image-generation provider options with Firecrawl web search and Gemini/llmman image generation, adds reasoning effort controls, and tightens request-parameter handling for workspace temperature and Generic OpenAI max_tokens.

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

01 **Generic OpenAI max_tokens omission via 0** NEW 75

Allows setting Generic OpenAI `max_tokens` to `0` to omit the parameter from the request entirely.

– Exact provider, field name, and value given v1.17.0

THINNER COVERAGE BELOW

02 **Workspace temperature applied to agent sessions** IMPROVED 55

Applies the workspace temperature setting to agent sessions and omits the parameter entirely when unset.

– Names setting and behaviour change, no exact config path v1.17.0

03 **Gemini and llmman image generation providers** NEW 45

Adds Gemini and llmman as new image generation providers, and generated images are now included in chat exports.

– Names both new providers and export behaviour, no setup steps v1.17.0

04 **Workspace settings autosave** IMPROVED 40

Autosaves workspace settings on edit, removing the need to manually submit changes.

– Explains before/after behaviour but no exact UI location v1.17.0

05 **Firecrawl as agent web-search provider** NEW 35

Adds Firecrawl as an agent web-search provider alongside existing options.

– Names the provider but no config or setup detail v1.17.0

06 **Keyboard navigation in provider search menus** IMPROVED

30

Adds keyboard navigation and autofocus to provider search menus.

— Thin UX note without specifics on scope v1.17.0

07 **Reasoning effort controls for supported models** NEW

25

Adds reasoning effort controls for supported models.

— No mechanism, UI path, or which models supported v1.17.0

08 **Time-of-day chat greeting rotation** NEW

25

Rotates empty chat greetings by time of day.

— Brief description, no configuration detail given v1.17.0

v1.17.0 NOTES ↗ LATEST STABLE

AnythingLLM v1.17.0 adds Firecrawl web-search, Gemini and llmman image generation, reasoning effort controls, and chat-greeting rotation.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.17.0 https://github.com/Mintplex-Labs/anything-llm.git
# already have the repo? check out this version:
$ git checkout v1.17.0
```

Adds Firecrawl as an agent web-search provider alongside existing options.

Adds Gemini as an image generation provider.

Adds llmman as an image generation provider.

Adds reasoning effort controls for supported models.

Includes generated images in chat exports.

+5 more

WAS THIS USEFUL? 👍 👎

◆ Local LLM Hardware

ASUS Ascent GX10

Feedback ⓘ price change · Oct 2, 2026

STORE ↗

ASUS's own GB10 Grace Blackwell desktop AI supercomputer — the only DGX Spark-class OEM twin with a scrapable live price, and it has climbed more than any other.

Ascent GX10 · 128 GB / 1 TB (gx10-1tb) list price moved from $6,999 to $0 (-100%) at `eshop.asus.com` on 2 Oct 2026.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Weaviate

Feedback ⓘ v1.39.8 NOTES ↗ RANK

Weaviate is an open-source vector database that stores both objects and vectors, allowing for the combination of vector search with structured filtering with the fault tolerance and scalability of a cloud-native database.

> Weaviate v1.39.8 adds a REST near-vector search endpoint and expands observability with license state visibility and new metric labels for consistency level and collection namespace.

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

01 **License state and monitoring label exposure** NEW 70

Exposes license state via `/v1/meta`, Prometheus metrics, and the startup log, and adds a `collection_namespace` label (alongside consistency level labeling) to usage metrics for finer-grained monitoring segmentation.

Check the current license state of a Weaviate node without tailing logs.

```
$ curl -s http://localhost:8080/v1/meta | jq '.license'
```

— Includes a runnable curl check for license state v1.39.8

THINNER COVERAGE BELOW

02 **REST near-vector search endpoint** NEW 55

Adds a `near-vector` REST search endpoint for vector similarity queries over the REST API, enabling vector similarity search without using the GraphQL API.

— Names endpoint but no request/response example v1.39.8

v1.39.8 NOTES ↗ LATEST STABLE

Weaviate v1.39.8 adds a REST near-vector search endpoint, license state exposure via `/v1`/meta, and new monitoring labels for consistency level and collection namespace.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.39.8
https://github.com/weaviate/weaviate.git
# already have the repo? check out this version:
$ git checkout v1.39.8
```

└──▷ **TRY IT**

Check the current license state of a Weaviate node without tailing logs.

```
$ curl -s http://localhost:8080/v1/meta | jq '.license'
```

Adds a `near-vector` REST search endpoint for vector similarity queries over the REST API.

Exposes license state via `/v1/meta`, Prometheus metrics, and the startup log.

Adds `collection_namespace` label to usage metrics for finer-grained monitoring segmentation.

WAS THIS USEFUL? 👍 👎

Volcengine OpenViking

Feedback ⓘ v0.4.23 NOTES ↗

RANK

OpenViking is a context database that manages agent memory, knowledge retrieval, and skills.

> OpenViking v0.4.23 overhauls retrieval to a single global vector recall with unified rerank, adds BM25 keyword search and Choice rerank mode, introduces ACL management tools, and reworks the installer and reindex pipeline alongside several breaking config and dependency changes.

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

01 **Reindex command overhaul** BREAKING 95

Adds a `force` flag to `reindex` and moves it to the RFV planner, making async reindex tasks recoverable; `queue_workers.reindex.max_concurrent` now defaults to `4`. Validation is also tightened: a non-recursive call such as `ov reindex viking://user/alice --mode semantic_and_vectors --recursive=false` now returns `INVALID_ARGUMENT` instead of silently rebuilding the whole namespace, so callers must target a specific resource, memory, or skill directory.

— Exact command, flag, error code and config default given v0.4.23

02 **Choice rerank mode in Jev** NEW 90

Adds `rerank.mode: "choice"` to enable Choice mode in Jev rerank, which requires `rerank.threshold: 0` and `min_score=0` for MCP calls; the default mode remains `noul`.

Enable Choice rerank mode so results are ranked by relative probability within the candidate pool — useful when you want softer, pool-relative scoring instead of absolute thresholds.

yaml

```yaml
rerank:
  mode: "choice"
  threshold: 0
```

— Exact config keys, requirements and default given v0.4.23

03 **Session auto-commit config merge** BREAKING 90

`memory.session_auto_commit.default_enabled` and `idle_enabled` are merged into a single `enabled` key (default `false`); old keys are ignored with a WARNING, so

any deployment that previously set them will have auto-commit silently disabled after upgrade — set `memory.session_auto_commit.enabled: true` to restore it.

— Exact config keys and the fix to restore old behavior v0.4.23

04 **BM25 keyword search** NEW 85

Adds `search_type="keywords"` to the `search` call for BM25 keyword retrieval, available across REST, MCP, CLI, and the Python, TypeScript and Go SDKs — useful for exact-term matching instead of vector similarity.

Run a BM25 keyword search from the CLI when exact-term matching is more useful than vector similarity.

```
$ ov search --search_type="keywords" "authentication bypass"
```

— Full surface list and runnable command given v0.4.23

05 **Retrieval overhaul to global vector recall** BREAKING 85

Retrieval changes to a single global vector recall with unified rerank; this removes directory recursion, parent score propagation, and hotness weighting, deleting the `retrieval.hotness_alpha` and `score_propagation_alpha` config keys (leftover entries are ignored with a WARNING). Result order and scores for the same query may differ from v0.4.22, so callers using `score_threshold` or score-based filters must recalibrate.

— Mechanism and before/after given, no runnable example v0.4.23

06 **Memory directory compilation skill** NEW 80

Adds `ov compile --skill memory` to deduplicate, merge, and normalize a memory directory in place, useful before a knowledge-base review or migration.

Deduplicate and consolidate a memory directory in place before a knowledge-base review or migration.

```
$ ov compile --skill memory
```

— Runnable command with usage example given v0.4.23

07 **Installer switches to release-channel-only npm install** 80

BREAKING

The installer now installs from the release channel only (no `git clone`), previews changes before applying them, and installs the `ov` CLI by default via `npm install -g @openviking/cli`, which requires npm on the machine. It resolves package downloads from whichever of `docs.openviking.net` and `docs.openviking.ai` responds first, and the old `--dist github`, `--source remote`, and `OPENVIKING_REPO_URL/REF/BRANCH` options are still accepted but now only print a notice instead of acting.

— Names install command and all legacy flags affected v0.4.23

---

08 **Scan rate limiting config change** BREAKING 80

`scan_batch_size` and `scan_batch_pause_seconds` are removed and replaced by `scan_rate_limit_files_per_second` (default `2.0`); `check_interval_seconds` default changes from `60` to `600` seconds.

— Exact old/new config keys and defaults given v0.4.23

---

09 **Health endpoint authentication error** BREAKING 70

`/health` now returns an authentication error when a request explicitly presents an expired or invalid API key; probes that send an invalid key should drop the credentials.

— Named endpoint and exact remediation for probes v0.4.23

---

10 **Gemini extras dependency bump** BREAKING 70

The `gemini` and `gemini-async` extras now require `google-genai>=1.39.0`; older SDKs raise `TypeError` in embed_async() and `AttributeError` in close().

— Exact version requirement and failure symptoms named v0.4.23

---

11 **ACL management tools and resource ACLs** NEW 65

Adds `list_users`, `list_groups`, `get_acl`, and `set_acl` MCP tools for managing access control, and extends the `write` and `add_resource` calls to accept an `acl` parameter for setting access control directly on resources.

— Lists all new ACL surfaces but no usage example v0.4.23

---

12 **ls and tree listing improvements** IMPROVED 65

Adds `include_abstract` and `include_overview` parameters to `ls`; adds a `directories_only` filter and L0/L1 content display to `tree`; and adds a `has_more` pagination flag to both `ls` and `tree` responses.

— Names all new ls/tree parameters but no usage example v0.4.23

THINNER COVERAGE BELOW

---

13 **Unified agent plugin capture filtering** IMPROVED 55

Agent plugin capture filtering is unified as sanitize → filter → truncate across pi, Codex, DSH, OpenCode, and Claude Code.

— Names pipeline stages and five integrations, no example v0.4.23

14 **Recall hooks forward exclusion and query filters** NEW 55

Claude Code and Codex recall hooks now forward `recallExcludeUris`; pi additionally supports `recallExcludeUris` and `recallQueryFilters`.

— Named fields and integrations but no usage example v0.4.23

15 **Event time-decay ranking parameter** NEW 45

Adds the `events_time_decay_protection` ranking parameter to `find` and `search` for event time-decay ordering of results.

— Named parameter but no mechanism detail or example v0.4.23

16 **MCP tool behavior annotations and error signaling** IMPROVED 45

MCP tools now advertise behavior annotations and return `isError: true` on whole-call failure.

— Names the field but no example of consuming it v0.4.23

17 **OIDC token support in ovcli.conf** NEW 35

Adds `oidc_token` support to `ovcli.conf` for authentication configuration.

— Named config key only, no further detail v0.4.23

18 **Kanban skill for task handoff** NEW 35

Adds the `ov-kanban` skill for structured task handoff.

— Bare mention of new skill, no mechanism given v0.4.23

19 **Studio account-level memory extraction rules** NEW 30

Studio adds account-level memory extraction rule editing.

— Bare feature name with no UI path or mechanism v0.4.23

20 **Bundled skills for pi and OpenCode** NEW 30

pi and OpenCode now ship with OpenViking skills bundled.

— Bare statement, no detail on which skills v0.4.23

21 **Observer reports vector metric and score scale** IMPROVED 25

Observer now reports the vector metric and score scale used for retrieval.

— Brief statement with no further detail v0.4.23

└──▷ BREAKING ON UPGRADE

! `memory.session_auto_commit.default_enabled` and `idle_enabled` are merged into `enabled` (default `false`); old keys are ignored with a WARNING, and any deployment that set them will have auto-commit silently disabled after upgrade — set `memory.session_auto_commit.enabled: true` to restore.

! `scan_batch_size` and `scan_batch_pause_seconds` are removed and replaced by `scan_rate_limit_files_per_second` (default `2.0`); `check_interval_seconds` default changes from `60` to `600` seconds.

! `retrieval.hotness_alpha` and `score_propagation_alpha` are removed; leftover config entries are ignored with a WARNING. Result order and scores for the same query may differ from v0.4.22 — callers using `score_threshold` or score-based filters must recalibrate.

! `ov reindex viking://user/alice --mode semantic_and_vectors --recursive=false` now returns `INVALID_ARGUMENT` instead of silently rebuilding the whole namespace; send the request to a specific resource, memory, or skill directory.

! `/health` now returns an authentication error when a request explicitly presents an expired or invalid API key; probes that send an invalid key should drop the credentials.

! The `gemini` and `gemini-async` extras now require `google-genai>=1.39.0`; older SDKs raise `TypeError` in embed_async() and `AttributeError` in close().

! The default installer no longer runs `git clone`; `--dist github`, `--source remote`, and `OPENVIKING_REPO_URL/REF/BRANCH` are still accepted but only print a notice. The `ov` CLI is now installed by default via `npm install -g @openviking/cli`, requiring npm on the machine.

v0.4.23 NOTES ↗ LATEST STABLE

OpenViking v0.4.23 adds BM25 keyword search, ACL tools, Choice rerank mode, kanban skill, and major retrieval/ranking overhaul.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.4.23
https://github.com/volcengine/OpenViking.git
# already have the repo? check out this version:
$ git checkout v0.4.23
```

└──▷ TRY IT

Run a BM25 keyword search from the CLI when exact-term matching is more useful than vector similarity.

```
$ ov search --search_type="keywords" "authentication bypass"
```

Enable Choice rerank mode so results are ranked by relative probability within the candidate pool — useful when you want softer, pool-relative scoring instead of absolute thresholds.

```yaml
rerank:
  mode: "choice"
  threshold: 0
```

Deduplicate and consolidate a memory directory in place before a knowledge-base review or migration.

```
$ ov compile --skill memory
```

Adds `search_type="keywords"` to `search` for BM25 keyword retrieval across REST, MCP, CLI, and the Python / TypeScript / Go SDKs.

Adds `rerank.mode: "choice"` to enable Choice mode in Jev rerank; requires `rerank.threshold: 0` and `min_score=0` for MCP calls; default stays `noul`.

Adds `events_time_decay_protection` ranking parameter to `find` and `search` for event time-decay ordering.

Adds `list_users`, `list_groups`, `get_acl`, and `set_acl` MCP tools; `write` and `add_resource` now accept an `acl` parameter.

Adds `oidc_token` support to `ovcli.conf`.

+15 more

└─▷ **BREAKING ON UPGRADE**

! `memory.session_auto_commit.default_enabled` and `idle_enabled` are merged into `enabled` (default `false`); old keys are ignored with a WARNING, and any deployment that set them will have auto-commit silently disabled after upgrade — set `memory.session_auto_commit.enabled: true` to restore.

! `scan_batch_size` and `scan_batch_pause_seconds` are removed and replaced by `scan_rate_limit_files_per_second` (default `2.0`); `check_interval_seconds` default changes from `60` to `600` seconds.

! `retrieval.hotness_alpha` and `score_propagation_alpha` are removed; leftover config entries are ignored with a WARNING. Result order and scores for the same query may differ from v0.4.22 — callers using `score_threshold` or score-based filters must recalibrate.

- ! `ov reindex viking://user/alice --mode semantic_and_vectors --recursive=false` now returns `INVALID_ARGUMENT` instead of silently rebuilding the whole namespace; send the request to a specific resource, memory, or skill directory.
- ! `/health` now returns an authentication error when a request explicitly presents an expired or invalid API key; probes that send an invalid key should drop the credentials.
- ! The `gemini` and `gemini-async` extras now require `google-genai>=1.39.0`; older SDKs raise `TypeError` in embed_async() and `AttributeError` in close().
- ! The default installer no longer runs `git clone`; `--dist github`, `--source remote`, and `OPENVIKING_REPO_URL/REF/BRANCH` are still accepted but only print a notice. The `ov` CLI is now installed by default via `npm install -g @openviking/cli`, requiring npm on the machine.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261002 NOTES ↗

RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith added a new thread/trace statistics endpoint and a faster data-export mode, standardized how Gateway-backed prompts are saved and run, and tightened dataset export permissions as a breaking change.

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

01 **Thread and trace stats endpoint** NEW 90

Adds `GET /v2/threads/stats` to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project, including on self-hosted ClickHouse deployments when SmithDB queries are disabled.

— Named endpoint with method and path, no usage example snapshot-20261002

02 **Skip rollup computation in data export** NEW 90

Data export requests gain an `is_rollup_disabled` boolean field (default `false`); setting it to `true` skips trace-level token and cost rollup computation to make exports faster, with root runs then carrying their own token/cost values instead of whole-trace totals. Applies only to exports reading from SmithDB.

— Named flag, default, and mechanism described product docs

03 **Standard manifest for Gateway prompts** IMPROVED 80

Prompts saved with a LangSmith Gateway model now use a standard `ChatOpenAI` manifest with the unified Gateway URL and a `LANGSMITH_GATEWAY_API_KEY` secret reference, making them pullable and runnable with standard LangChain SDKs.

— Names manifest format and secret variable, no runnable example snapshot-20261002

THINNER COVERAGE BELOW

04 **Application resource tag value** NEW 45

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

— UI path given, little further mechanism snapshot-20261002

05 **Deep Agent missing connection guidance** IMPROVED 35

Managed Deep Agent channel replies now name a missing workspace connection and show how an agent developer can configure it.

— Thin description, no concrete config shown snapshot-20261002

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read AND download permissions; existing API clients that have only dataset read permission will receive a forbidden response instead of the file.

snapshot-20261002 NOTES ↗ LATEST STABLE

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

└──▷ **USE IT**

Give self-hosted operators an earlier heads-up before license expiry — e.g. 30 days — so the warning banner appears with enough lead time to renew.

.env

```
LICENSE_EXPIRATION_WARNING_DAYS=30
```

Tag a custom OpenAI-compatible model in LLM Gateway so its traces carry provider metadata and match the correct pricing tier.

json

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

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

Prompts saved with a LangSmith Gateway model now use a standard `ChatOpenAI` manifest with the unified Gateway URL and a `LANGSMITH_GATEWAY_API_KEY` secret reference, making them pullable and runnable with standard LangChain SDKs.

Managed Deep Agent channel replies now name a missing workspace connection and show how an agent developer can configure it.

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read AND download permissions; existing API clients that have only dataset read permission will receive a forbidden response instead of the file.

WAS THIS USEFUL?  

UK AI Security Institute Inspect AI

Feedback ⓘ

0.3.275 CODE ↗ RANK

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

> Inspect AI 0.3.275 improves Gemini model support when accessed through LiteLLM Proxy, adding reasoning-effort control and more robust tool-call handling.

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

01 **Gemini support via LiteLLM Proxy** IMPROVED 67

Adds `reasoning_effort` support for Gemini models accessed via LiteLLM Proxy, mapping the setting to the levels or thinking budgets used by the native Google provider. LiteLLM Proxy also now preserves Gemini thought signatures in tool calls when `model_name` does not contain `gemini`, carries the function-calling hint on requests with tools, and retries malformed function-call turns with a corrective exchange.

— Names config key and condition but no runnable example given. 0.3.275

0.3.275 CODE ↗ LATEST STABLE

Inspect AI 0.3.275 adds Gemini `reasoning_effort` and improved Gemini tool-call handling via LiteLLM Proxy.

└──▷ **GET THIS VERSION**

```
$ git clone --branch 0.3.275
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.275
```

Adds `reasoning_effort` support for Gemini models via LiteLLM Proxy, mapped to the levels or thinking budgets the native Google provider uses.

LiteLLM Proxy now preserves Gemini thought signatures in tool calls when `model_name` does not contain `gemini`, carries the function-calling hint on requests with tools, and retries malformed function-call turns with a corrective exchange.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261002 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust's biggest additions this window are span-customizer APIs for redacting or modifying spans across its Python, TypeScript, Go, and Java SDKs, SQL-backed dashboard charts that plot time series, top lists, and tables, and async workflow evaluations in TypeScript — alongside breaking changes to OpenAI attachment capture, durable evals, and Java span tracer fields, plus SCIM-based org provisioning.

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

01 Extended OpenAI instrumentation with attachment capture control

93

BREAKING

Extends TypeScript SDK v3.35.0 instrumentation to cover OpenAI audio APIs (speech, transcription, translation), image generation, image editing, and video generation. As part of this, base64 image and file inputs to OpenAI Chat Completions calls are now recorded as `<omitted>` by default instead of being uploaded as attachments automatically; set `BRAINTRUST_CAPTURE_ATTACHMENTS=true` to restore the previous behavior of uploading them for inspection in traces.

Upload OpenAI image generation outputs as attachments for inspection in Braintrust traces, rather than having them omitted.

```
$ export BRAINTRUST_CAPTURE_ATTACHMENTS=true
```

— Exact env var and command to restore behavior product docs

02 SQL-backed dashboard charts NEW

91

Adds SQL charts to dashboards (public preview, Pro/Enterprise only, requires data plane v2.1.1+): build a chart from a raw SQL query or a visual builder that edits the same underlying query, and plot results as time series, top lists, or tables. The dashboard's timeframe, filters, and grouping are automatically injected into the query at render time, making a single SQL chart reusable across any dashboard it is pasted into.

Create a SQL chart on a dashboard to track error rates over time, using the dashboard's timeframe filter automatically.

> 📍 In Braintrust, open a dashboard › Add chart › SQL chart. Enter your query (e.g. SELECT date_trunc('hour', timestamp) AS t, COUNT(*) AS errors FROM logs WHERE

level = `error` GROUP BY t ORDER BY t). Switch to the visual builder to refine grouping, then set the plot type to `Time series`. Save and paste the chart into any other dashboard — the dashboard's timeframe and filters are applied automatically at render time.

— Full UI steps and version requirement given `snapshot-20261002`

---

03 **Span customizers for redacting spans across SDKs** NEW

85

Adds span customizer APIs across four SDKs to modify or redact spans before upload: set_span_customizers() (Python v0.43.0+), configureInstrumentation() with `spanCustomizers` (TypeScript v3.35.0+), `braintrust.WithSpanCustomizers` (Go v0.16.0+), and addSpanCustomizer() / `BraintrustConfig.Builder` (Java v0.3.25+) to register `SpanCustomizer` callbacks. Built on these, the TypeScript and Python SDKs add sensitive-data redaction via masking functions and span customizers that strip PII from traces before upload, while the Go (v0.16.0) and Java (v0.3.25) SDKs run span customizers on every exported span for the same purpose.

— Names exact API per SDK but no code sample `product docs`

---

04 **Workflow evaluations for async provider batch APIs** BREAKING

85

Adds defineWorkflowEval() to the TypeScript SDK (v3.33.0+) for submitting eval tasks and scorers to asynchronous provider batch APIs, collecting results via polling or webhooks. This replaces defineDurableEval() and its processBatchResult() method, along with `BatchTask`, `BatchScorer`, `DurableEvalStore`, `DurableEvalMemoryStore`, and `DurableEvalRedisStore`, all removed in TypeScript SDK v3.35.0; existing users must migrate to defineWorkflowEval().

— Migration path given but no code sample `product docs`

---

05 **Tracer field on Java SDK spans** BREAKING

78

Adds the `tracer` field to Java SDK spans (v0.3.25+), populated with the name of the tracer that created the span (e.g. `aws-bedrock`, `genai`, `langchain4j`, `springai`) instead of always being `braintrust-java`; queries that filter on `braintrust-java` must be updated.

— Migration note given but no example query `product docs`

THINNER COVERAGE BELOW

---

06 **Chart data-point actions on dashboards** NEW

50

Clicking a data point on a dashboard chart opens a menu to open matching traces in a side panel, explore rows in the SQL sandbox, or filter the dashboard to the clicked

series.

– UI path given, no deeper mechanism product docs

07 **Extended tracing capture fields in Python SDK** NEW 50

Adds `stop_details`, `compaction` request parameters, `speed`, and `fallback_credit` usage fields to tracing capture in Python v0.43.0.

– Named fields but no usage example product docs

08 **Recent SQL filter suggestions in search** IMPROVED 45

Search dropdown now surfaces recent SQL filters matching the typed prefix, enabling reuse of previous filters without clearing the field.

– Describes behavior, no named field or key product docs

09 **UTC datetime normalization at index time** IMPROVED 45

Datetime fields in trace data are now normalized to UTC at index time by Brainstore, with documentation explaining how to preserve the original timezone offset.

– Behavior change noted, no config key given product docs

10 **Retry support for failed Loop automation runs** NEW 40

Adds retry support for failed Loop automation runs, allowing recovery from transient failures without reconfiguring the automation.

– No UI path or command named product docs

11 **Collapsible guide sections in trace view** IMPROVED 40

Trace view now shows guide sections as collapsible timeline groups, preserving span order, bars, and hierarchy.

– UI change described, no navigation path product docs

12 **SCIM group-based provisioning** NEW 35

Supports SCIM group-based provisioning to automatically derive organization membership and optionally permission group membership from identity provider groups.

– No setup steps or endpoint named product docs

└──▷ **BREAKING ON UPGRADE**

! TypeScript SDK v3.35.0: Base64 image and file inputs to OpenAI Chat Completions calls are now recorded as `<omitted>` by default unless `BRAINTRUST_CAPTURE_ATTACHMENTS` is set to true; earlier versions uploaded them as attachments automatically.

! TypeScript SDK v3.35.0: defineDurableEval() and its processBatchResult() method, along with `BatchTask`, `BatchScorer`, `DurableEvalStore`, `DurableEvalMemoryStore`, and `DurableEvalRedisStore`, are removed; migrate to defineWorkflowEval().

! Java SDK v0.3.25+: The `tracer` field on integration spans now holds the actual tracer name (e.g. `aws-bedrock`, `genai`, `langchain4j`, `springai`) instead of always being `braintrust-java`; queries that filter on `braintrust-java` must be updated.

`snapshot-20261002` NOTES ↗ **LATEST** STABLE

Braintrust adds SQL-driven dashboard charts with time-series, top-list, and table plots in public preview.

└──▷ **HOW TO FIND IT**

Create a SQL chart on a dashboard to track error rates over time, using the dashboard's timeframe filter automatically.

📍 In Braintrust, open a dashboard › Add chart › SQL chart. Enter your query (e.g. SELECT date_trunc('hour', timestamp) AS t, COUNT(*) AS errors FROM logs WHERE level = `error` GROUP BY t ORDER BY t). Switch to the visual builder to refine grouping, then set the plot type to `Time series`. Save and paste the chart into any other dashboard — the dashboard's timeframe and filters are applied automatically at render time.

Adds SQL charts (public preview, Pro/Enterprise only, requires data plane v2.1.1+): build dashboard charts from a raw SQL query or a visual query builder that edits the same underlying query.

Supports three plot types for SQL chart results: time series, top lists, and tables.

Automatically injects the dashboard's timeframe, filters, and grouping into the SQL query at render time, making a single SQL chart reusable across any dashboard it is pasted into.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.88 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik 2.2.88 focuses on annotation queue workflows, adds static custom headers for OpenAI and OpenRouter, simplifies MCP onboarding, and improves the prompt comparison view.

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

01 **Simplified opik mcp configure onboarding** IMPROVED 50

Simplifies the `opik mcp configure` onboarding flow for faster MCP setup.

– Names exact command but no detail on what changed 2.2.88

02 **Static custom headers for OpenAI and OpenRouter** NEW 42

Adds support for static custom headers on OpenAI and OpenRouter providers, configurable in the UI.

– Names providers and UI location but no field names 2.2.88

03 **Annotation queue visibility and automation** IMPROVED 37

Traces and threads tables now show annotation queue membership directly in the UI, and annotation queue automation is turned on by default.

– Describes what changed but no mechanism or config detail 2.2.88

04 **Prompt comparison view improvements** IMPROVED 35

Renders compared prompts as messages and stores prompt templates within the prompt comparison view.

– States behaviour change but lacks further specifics 2.2.88

2.2.88 NOTES ↗ LATEST STABLE

Opik 2.2.88 adds annotation queue visibility in traces/threads tables, static custom headers for OpenAI/OpenRouter, and a simplified `opik mcp configure` onboarding.

└──▷ **GET THIS VERSION**

```
$ git clone --branch 2.2.88 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.88
```

Simplifies the `opik mcp configure` onboarding flow for faster MCP setup.

Allows static custom headers on OpenAI and OpenRouter providers in the UI.

Shows annotation queue membership directly in traces and threads tables.

Turns annotation queue automation on by default.

Renders compare prompts as messages and stores prompt templates in the prompt comparison view.

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.50.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse's v4.50.0 release expands the evaluations engine with new facet types, built-in protection, and rule idle time controls, and adds seven-day execution health visibility to evaluator status.

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

01 **Seven-day execution health in evaluator status** IMPROVED

30

Evaluator status now shows seven-day execution health, giving an at-a-glance view of evaluator reliability.

— States what it shows but not where or how to view it v4.50.0

02 **Evaluations engine gains facet type, protection, and idle time**

NEW

27

The evaluations engine adds a `facet type`, built-in protection, and rule idle time controls.

— Names three additions but no mechanism or usage detail v4.50.0

v4.50.0 NOTES ↗ LATEST STABLE

Langfuse v4.50.0 adds evaluator facet types, built-in protection, rule idle time, and seven-day execution health visibility.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v4.50.0
https://github.com/langfuse/langfuse.git
# already have the repo? check out this version:
$ git checkout v4.50.0
```

Adds facet type, built-in protection, and rule idle time to the evaluations engine.

Shows seven-day execution health in evaluator status for at-a-glance evaluator reliability.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

Giskard

Feedback ⓘ v3.0.1 NOTES ↗ RANK

Giskard tests and evaluates LLM applications for performance, bias, and security issues.

> Giskard's latest releases added SOM support for LLM judges, allowed model strings to be passed directly to the default generator, and introduced an agent quality gate for contributors.

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

01 **Agent quality gate for contributions** NEW 60

Adds agent quality rails with a per-file lint/typecheck gate and a `uv` guard for pre-commit hooks.

— Names the specific gate and guard mechanism added giskard-core/v1.0.2

THINNER COVERAGE BELOW

02 **Model strings in set_default_generator** IMPROVED 55

`set_default_generator` now accepts model strings directly, letting callers specify a generator by name rather than constructing an object.

— Names the exact function changed and new calling convention v3.0.1

03 **SOM support for LLM judges** NEW 20

Giskard adds SOM (Self-Organizing Map) support for judges, mentioned across both the core and main package releases, though no further mechanism or configuration detail is given.

— Named only in summaries, no mechanism or usage detail given giskard-core/v1.0.2 v3.0.1

v3.0.1 NOTES ↗ LATEST STABLE

Giskard v3.0.1 adds SOM support for judges and accepts model strings in `set_default_generator`.

└──▷ GET THIS VERSION

```
$ git clone --branch v3.0.1 https://github.com/Giskard-AI/giskard-oss.git
# already have the repo? check out this version:
$ git checkout v3.0.1
```

Accepts model strings directly in `set_default_generator`, letting callers specify a generator by name rather than constructing an object.

giskard-core/v1.0.2 NOTES ↗ STABLE

Giskard core v1.0.2 adds SOM (Self-Organizing Map) support for LLM judges and a per-file lint/typecheck agent quality gate.

└──▷ GET THIS VERSION

```
$ git clone --branch giskard-core/v1.0.2 https://github.com/Giskard-AI/giskard-oss.git
# already have the repo? check out this version:
$ git checkout giskard-core/v1.0.2
```

Adds agent quality rails with a per-file lint/typecheck gate and `uv` guard for pre-commit hooks.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

DeepSeek V4 Flash 0731 Feedback ⓘ

price cut · Oct 1, 2026 MODEL ↗

DeepSeek V4 Flash 0731 is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4-flash): $0.047 → $0.042 in, $0.0941 → $0.084 out per million tokens (-11%), first seen 1 Oct 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V4 Pro 0813

Feedback ⓘ price cut · Oct 1, 2026

MODEL ↗

DeepSeek V4 Pro 0813 is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4-pro): $0.6116 → $0.2088 in, $1.2232 → $0.4176 out per million tokens (-66%), first seen 1 Oct 2026.

WAS THIS USEFUL?  

Black Forest Labs FLUX

Feedback ⓘ FLUX 3 Image NOTES ↗

RANK

FLUX is Black Forest Labs' family of visual generation and editing models.

> FLUX 3 Image launches as a single endpoint unifying generation, editing, and multi-reference composition up to 4K, adding prompt-driven layout control, pixel-exact local edits, and support for up to ten reference images in one request.

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

01 **Bounding-box layout control via prompt field** NEW 80

Adds bounding-box layout control via a JSON list appended to the `prompt` field — no separate parameter — to place headlines, faces, grid panels, or any element at pixel-exact positions.

— Names exact field and JSON mechanism but no sample payload shown FLUX 3 Image

02 **Unified generate-or-edit endpoint** NEW 60

Exposes a single endpoint that infers generate-or-edit intent directly from the `prompt` contents, with no `mode` field required, covering generation, editing, and multi-reference composition up to 4K resolution.

— Names the endpoint behavior and dropped `mode` field, no path given FLUX 3 Image

THINNER COVERAGE BELOW

03 **Pixel-exact local edits in a single request** NEW 50

Supports pixel-exact local edits — recolor, replace, move, resize, or remove multiple elements — in a single request by marking the target region.

— Describes edit types but not how regions are marked FLUX 3 Image

04 **Multi-reference composition with up to ten images** NEW 40

Accepts up to ten named image references in one request for restyling or compositing new images.

— Gives a concrete limit but no field name or request shape FLUX 3 Image

FLUX 3 Image NOTES ↗ LATEST STABLE

> FLUX 3 Image launches a single endpoint for generation, editing, and multi-reference composition up to 4K.

> Adds bounding-box layout control via a JSON list appended to the `prompt` field — no separate parameter — to place headlines, faces, grid panels, or any element at pixel-exact positions.
>
> Supports pixel-exact local edits (recolor, replace, move, resize, or remove multiple elements) in a single request by marking the target region.
>
> Accepts up to ten named image references in one request for restyling or compositing new images.
>
> Exposes a single endpoint that infers generate-or-edit intent from the `prompt` contents, with no `mode` field required.

WAS THIS USEFUL? 👍 👎

Moonshot AI Kimi K2.6 Feedback ⓘ price cut · Oct 1, 2026

MODEL ↗

Kimi K2.6 is a multimodal model from Moonshot AI.

Price cut on OpenRouter (moonshotai/kimi-k2.6): $0.65 → $0.4341 in, $3.41 → $1.828 out per million tokens (-46%), first seen 1 Oct 2026.

WAS THIS USEFUL? 👍 👎

MiniMax M1 Feedback ⓘ price increase · Oct 1, 2026 MODEL ↗

MiniMax M1 is a chat model from MiniMax.

Price increase on OpenRouter (minimax/minimax-m1): $0.40 → $0.55 in, $2.20 → $2.20 out per million tokens (+38%), first seen 1 Oct 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 2, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue