THE AI TOOLCHAIN

NO. 100

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED NOVEMBER 24, 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

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 346 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 72 of 75 tools

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

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

BUILD

vim ? help

Run a crew of agents under one coordinator instead of babysitting each session

Kiro Crew 0.13.0 adds Conductor mode for orchestrating worker agents, a rebuilt browser agent with replay, and Memory overhaul with Ollama embeddings; OpenCode ships a Code Mode tool catalog covering browser automation, subagents and file editing. Kiro's Claude Fable 5.2 Preview brings a 1M context window and subagent coordination for cross-repo work, which is the case single-session agents keep failing on.

Kiro Crew · OpenCode · Kiro

GOVERN

Stop a sandbox backup from shipping live credentials

NemoClaw scans for credential material before publishing a backup and rejects archives still containing SQLite WAL/SHM/journal files — the half-written state that leaks rows a clean export would not. OpenViking's new `ov privacy run` executes subprocesses with secrets redacted, so shelling out from an agent stops being the place tokens escape into logs.

NemoClaw · OpenViking

DEPLOY

Catch a deployment pointed at the wrong config store before it serves answers

LightRAG v1.5.8 adds a selectable configuration-storage backend plus an anchor file binding a deployment to its backend, and enforces an embedding baseline — so a restart against a mismatched store or a changed embedding model fails loudly instead of returning quietly wrong retrievals. Four offline maintenance CLI tools cover the repair side.

lightrag

Does **Opus 5** read better? 👍 👎

BUILD

◆ AI Coding Agents

SST OpenCode

Feedback ⓘ updated · Nov 24, 2026 NOTES ↗ RANK

OpenCode is an open-source AI coding agent for terminal-based development tasks.

> OpenCode shipped a major expansion of its agent platform this window: a full Code Mode tool catalog (browser automation, subagents, image generation, file editing, skills, web access), a layered permissions/policies system that reaches down to MDM-managed preferences, and an agents/commands/plugins/MCP framework for building custom workflows — alongside provider, model, formatter, reference, theme, and networking configuration overhauls.

└──▷ **WHAT SHIPPED · 25 FEATURES** most completely described first ? what's the number?

01 **MCP server management: add, pool, OAuth, timeouts** Not checked yet

NEW 98

Add a remote MCP server with `opencode mcp add <name> --url <url>` (optionally `--global`), or a local stdio server with `opencode mcp add <name> -- <command> [args]` using repeated `--env` flags; pass secret headers inline with `--header` and `{env:NAME}`. Configure per-server options under `mcp.servers.<name>` in JSONC: `cwd`, `env`, `disabled`, `codemode` (to expose tools directly on the provider's native tool list instead of through Code Mode), `protocol` (`legacy` default up to 2025-11-25, `auto` to probe via `server/discover`, or `2026-07-28`), and `pool: true` to share one connection across directories (requires the 2026-07-28 revision and sets `protocol` to `auto` if unset). Set per-phase `timeout` overrides for `startup` (default 30s), `catalog` (default 30s), and `execution` (default 12h) under a server's `timeout` object or globally under `mcp.timeout`. Configure OAuth with snake_case `oauth.client_id`, `oauth.client_secret`, `oauth.callback_port`, `oauth.redirect_uri`, and `oauth.auth_server_metadata_url`, supporting Client ID Metadata Document (CIMD) and dynamic client registration (DCR) flows with automatic fallback, or disable OAuth entirely with `oauth: false` for API-key-only servers; remove stored credentials and force re-authentication with `opencode mcp auth`. Control tool/resource visibility with permission actions matching `<server>_<tool>` and `opencode_list_mcp_resources` / `opencode_read_mcp_resource` scoped by server name or `<server>:<uri>`. Server initialization instructions are injected into the system prompt inside `<mcp_instructions>` tags (omitted only when all a server's tools are denied); servers can request user input mid-call via MCP elicitation, surfaced as an in-app form or URL link; and OpenCode sends the invoking session ID in `CallToolRequest.params._meta["ai.opencode/sessionID"]` for correlation.

Add a project-scoped remote MCP server that authenticates with an API key header, keeping the secret out of config files.

```
$ opencode mcp add context7 --url https://mcp.context7.com/mcp --
header 'CONTEXT7_API_KEY={env:CONTEXT7_API_KEY}'
```

Share one Cloudflare MCP connection across all open project directories instead of opening a separate connection per directory.

```json
{
  "mcp": {
    "servers": {
      "cloudflare": {
        "url": "https://mcp.cloudflare.com/mcp",
        "pool": true,
        "protocol": "2026-07-28"
      }
    }
  }
}
```

Add a remote MCP server globally so it is available in every project, authenticating via an API key header rather than OAuth.

```
$ opencode mcp add context7 --global --url
https://mcp.context7.com/mcp --header 'CONTEXT7_API_KEY=
{env:CONTEXT7_API_KEY}'
```

Share one Cloudflare MCP connection across all open directories instead of opening one per directory, using the 2026-07-28 protocol revision.

```json
{
  "mcp": {
    "servers": {
      "cloudflare": {
        "url": "https://mcp.cloudflare.com/mcp",
        "pool": true,
        "protocol": "2026-07-28"
      }
    }
  }
}
```

Cap how long a slow local MCP server can take to list its tools, while keeping the default 12-hour execution timeout for long-running tool calls.

```json
{
  "mcp": {
    "servers": {
      "slow-tools": {
        "command": "npx",
        "args": ["-y", "slow-tools-mcp"],
        "timeout": {
          "catalog": 5000
        }
      }
    }
  }
}
```

— Every CLI flag, config key, and default timeout named with runnable examples.

product docs

---

02 **Permissions rule system for agent actions** Not checked yet NEW

96

Define ordered `permissions` rules in `opencode.jsonc`, each with `action`, `resource`, and `effect` (`allow`, `ask`, or `deny`), scoped globally or per agent via `agents.<id>.permissions` (agent rules append after global ones). Wildcard shell resources with a trailing `*` to match a command and its arguments (e.g. `git status *`); `~/` and `$HOME/` prefixes are expanded at load time for `read`, `edit`, and `glob` resources (not for shell). Add an `external` action rule (`resource: *`, `effect: ask`) to prompt before any file access outside the active Location, then narrow with `allow` / `deny` rules for specific directories. Built-in tool actions include `shell` (scanner-produced command string, scanned across Bash, Zsh, POSIX `sh`, and PowerShell, with malformed or unanalyzable commands erroring rather than executing), `skill`, `subagent`, `fetch`, `web_search`, `experimental_image_generate`, `external`, and `<server>_<tool>` for MCP tools (plus `opencode_list_mcp_resources` and `opencode_read_mcp_resource`). Built-in agents (`plan`, `general`, `explore`, `title`, `summary`, `compaction`) each start from a base policy that custom agents also inherit. When prompted, clients can reply Allow once, Allow always (saves a durable project-scoped `allow` rule), or Deny; saved approvals are stored as `shell` prefix patterns and never override a configured `deny`.

Ask before any shell command, silently allow routine git inspection, and block all git pushes — the most common agent permission setup.

```json
// opencode.jsonc
{
  "permissions": [
    { "action": "shell", "resource": "*",           "effect":
"ask"   },
    { "action": "shell", "resource": "git status *", "effect":
"allow" },
    { "action": "shell", "resource": "git diff *",   "effect":
"allow" },
    { "action": "shell", "resource": "git push *",   "effect":
"deny"  }
  ]
}
```

Restrict a custom `reviewer` subagent to read-only access while keeping global shell rules active for other agents.

```json
// opencode.jsonc
{
  "permissions": [
    { "action": "shell", "resource": "*",           "effect":
"ask"   },
    { "action": "shell", "resource": "git status *", "effect":
"allow" }
  ],
  "agents": {
    "reviewer": {
      "description": "Review code without changing it",
      "subagent": true,
      "permissions": [
        { "action": "edit", "resource": "*", "effect": "deny" }
      ]
    }
  }
}
```

Block all shell commands by default while allowing routine Git inspection without prompting — a safe baseline for an automated review agent.

```json
// opencode.jsonc
{
  "permissions": [
    { "action": "shell", "resource": "*", "effect": "deny" },
    { "action": "shell", "resource": "git status *", "effect":
"allow" },
    { "action": "shell", "resource": "git diff *", "effect":
"allow" }
  ]
}
```

Prompt before the agent reads or writes anything outside the current project, then explicitly allow a shared reference directory.

```json
// opencode.jsonc
{
  "permissions": [
    { "action": "external", "resource": "*", "effect": "ask" },
    { "action": "read",     "resource": "~/projects/reference/*",
"effect": "allow" }
  ]
}
```

— Exact rule schema, wildcard syntax, and action names with runnable config examples.

product docs

---

03 **Policies framework for hard, hierarchical restrictions**

Not checked yet

96

BREAKING

Author binary, never-prompting `policies` statements in `opencode.jsonc` that can only tighten what permissions/providers would otherwise allow: `integration.connect` and `integration.use` control available AI providers by provider ID (e.g. `anthropic`, `opencode`); `tool.use` deny statements use `<action>:<value>` resources like `shell:git push *`, `edit:*.env`, or `webfetch:*` (a denied check returns Blocked by configuration policy and overrides saved Allow always approvals); `mcp.use` controls MCP server availability by name; `skill.use` controls which skills are offered; `plugin.use` controls which plugins load by package name or path; and `websearch.use` controls which `websearch` providers (`opencode`, `exa`, `firecrawl`, `parallel`, `tavily`, `tinyfish`) are available. Policies are evaluated in a six-level authority hierarchy — macOS managed preferences at `/Library/Managed Preferences/ai.opencode.managed.plist`

ranked highest, then system opencode.json(c) at `/Library/Application Support/opencode`, `/etc/opencode`, or `%ProgramData%\opencode`, then a connected OpenCode Console workspace, then global ~/.config/opencode/opencode.json(c), then direct and project `opencode.jsonc` files. The deprecated `autoapprove` key still works but merges into `policies`. Before client 2.2.3, a bare `integration.use` deny also blocked MCP servers, skills, and plugins via `mcp:`, `skill:`, or `plugin:` prefixes; this no longer applies, so replace it with explicit `mcp.use`, `skill.use`, and `plugin.use` statements — clients older than 2.2.3 ignore `integration.connect` / `mcp.use` / `skill.use` / `plugin.use`, and clients older than 2.2.6 ignore `websearch.use`.

Lock a developer machine to a single managed Anthropic key and block all shell commands except `git status`, so the agent cannot exfiltrate data or run arbitrary commands.

```json
// ~/.config/opencode/opencode.jsonc
{
  "policies": [
    { "action": "integration.use", "resource": "*", "effect":
"deny" },
    { "action": "integration.use", "resource": "anthropic",
"effect": "allow" },
    { "action": "tool.use", "resource": "shell:*", "effect":
"deny" },
    { "action": "tool.use", "resource": "shell:git status *",
"effect": "allow" }
  ]
}
```

Enforce an MDM profile that allows only the company Console workspace and blocks every other provider, MCP server outside an approved list, and all web search except the Console-provided one.

```json
// /Library/Managed Preferences/ai.opencode.managed.plist
(deployed via MDM)
{
  "policies": [
    { "action": "integration.connect", "resource": "*", "effect":
"deny" },
    { "action": "integration.connect", "resource": "opencode",
"effect": "allow" },
    { "action": "integration.use", "resource": "*", "effect":
"deny" },
    { "action": "integration.use", "resource":
"opencode#org_01M3CAP8H43Z49CER6NT0Y3ZF1", "effect": "allow" },
    { "action": "mcp.use", "resource": "*", "effect": "deny" },
    { "action": "mcp.use", "resource": "github", "effect":
"allow" },
    { "action": "websearch.use", "resource": "*", "effect":
"deny" },
    { "action": "websearch.use", "resource": "opencode",
"effect": "allow" }
  ]
}
```

Prevent the agent from reading SSH keys or touching `.env` files anywhere in the project, as a project-level guardrail checked in to the repo.

```json
// .opencode/opencode.jsonc
{
  "policies": [
    { "action": "tool.use", "resource": "read:*/.ssh/*",
"effect": "deny" },
    { "action": "tool.use", "resource": "edit:*.env", "effect":
"deny" },
    { "action": "tool.use", "resource": "webfetch:*", "effect":
"deny" }
  ]
}
```

Lock a managed workstation to only the company's Console workspace, blocking all other providers and free-tier access.

```json
// ~/.config/opencode/opencode.jsonc
{
  "policies": [
    { "action": "integration.connect", "effect": "deny",
"resource": "*" },
    { "action": "integration.connect", "effect": "allow",
"resource": "opencode" },
    { "action": "integration.use", "effect": "deny", "resource":
"*" },
    { "action": "integration.use", "effect": "allow", "resource":
"opencode#org_01M3CAP8H43Z49CER6NT0Y3ZF1" }
  ]
}
```

Block all shell commands except `git status` to prevent the agent from running arbitrary shell commands in a project.

```json
// .opencode/opencode.jsonc
{
  "policies": [
    { "action": "tool.use", "effect": "deny", "resource":
"shell:*" },
    { "action": "tool.use", "effect": "allow", "resource":
"shell:git status *" }
  ]
}
```

Restrict web search to only the Console-managed OpenCode Web Search provider, disabling all third-party providers like Exa and Tavily.

```json
// ~/.config/opencode/opencode.jsonc
{
  "policies": [
    { "action": "websearch.use", "effect": "deny", "resource":
"*" },
    { "action": "websearch.use", "effect": "allow", "resource":
"opencode" }
  ]
}
```

Hard-block `sudo` across all projects at the org level, overriding anything a repository's own config would allow.

```json
// opencode.jsonc (global or console-managed)
{
  "policies": [
    { "action": "tool.use", "resource": "shell:sudo *", "effect":
"deny" }
  ]
}
```

Hard-deny `sudo` across the whole organisation so a repository's own permissions cannot re-enable it.

```json
// opencode.jsonc
{
  "policies": [
    { "action": "tool.use", "resource": "shell:sudo *", "effect":
"deny" }
  ]
}
```

— Full statement types, authority order, and version-gated breaking changes all named.

product docs

04 **Agent definition system with per-agent permissions**

Not checked yet NEW 96

Define reusable named agents in Markdown at `~/.config/opencode/agents/<name>.md` (global) or `.opencode/agents/<name>.md` (project), with frontmatter for `model`, `instructions`, `mode`, `system`, `permissions`, and display details including a six-digit hex color, and the document body as the system prompt; nested files namespace IDs (e.g. `.opencode/agents/team/reviewer.md` becomes `team/reviewer`). Set the default primary agent with `default_agent` in `opencode.jsonc`. Control where an agent runs with `role` (primary-only, subagent-only, or `general`). Restrict capabilities with an ordered `permissions` list of `action` / `resource` / `effect` rules (last match wins) covering shell, edit tools, child-agent launch, local discovery, web tools, and skill loading. Cap steps per agent with `steps` in frontmatter or config; on the final step OpenCode removes tools and asks for a text summary. Hide an agent from listings with `hidden: true`, or remove it entirely with `disabled: true`. Overlay per-agent HTTP `headers` and JSON `body` fields via the `request` map. Specify a model with `provider/model#variant` shorthand or expanded

`providerID` / `modelID` / `variant` keys. V1 frontmatter fields (including the legacy top-level `maxSteps`) are deprecated — files using them still load, but new V2 configuration should not use them.

Create a project-scoped read-only code-review subagent that the primary agent can launch by name, with shell access denied.

yaml

```yaml
# .opencode/agents/reviewer.md
---
description: Reviews changes for correctness and regressions
model: anthropic/claude-sonnet-5-5#high
permissions:
  - action: "*"
    resource: "*"
    effect: deny
  - action: shell
    resource: "*"
    effect: deny
---
Review the current changes. List findings in severity order with
file and line references.
```

Override the built-in coding agent to block all shell commands organisation-wide by shadowing its ID in the global config.

json

```json
{
  "agents": {
    "coder": {
      "permissions": [
        { "action": "shell", "resource": "*", "effect": "deny" }
      ]
    }
  }
}
```

Create a read-only code-review subagent that the primary agent can launch by name, scoped to source files only.

```yaml
# .opencode/agents/reviewer.md
---
description: Reviews changes for correctness and regressions
model: anthropic/claude-sonnet-5-5#high
permissions:
  - action: "*"
    resource: "*"
    effect: deny
  - action: read
    resource: "src/**"
    effect: allow
steps: 20
---
Review the current changes. List findings in severity order with
file and line references.
```

Override the built-in default coding agent to block shell access by default, using the same agent ID in `opencode.jsonc`.

```json
{
  "agents": {
    "coder": {
      "permissions": [
        { "action": "shell", "resource": "*", "effect": "deny" }
      ]
    }
  }
}
```

— File paths, frontmatter keys, and deprecation named with runnable examples.

product docs

---

05 **Custom and OpenAI-compatible provider configuration**

Not checked yet NEW 96

Add any OpenAI-compatible provider under the `providers` key in `opencode.jsonc` by setting `package` to `@opencode/ai/providers/openai-compatible`, `env` for the credential variable, and `settings.baseURL` for the endpoint; route a built-in catalog provider through a proxy by overriding its `settings.baseURL`. Native `package` options include `@opencode/ai/providers/openai`, `-anthropic`, `-google`, `-google-vertex`, `-azure`, `-amazon-bedrock`, `-openrouter`, and `-xai` (plus any npm package or `file://` URL); add or override models on a provider under its `models` map with `modelID`, `name`, `family`, `type`, `releaseDate`,

`capabilities`, `compatibility`, `variants`, `cost`, `limits`, and `disabled`. Customize provider and model requests with `settings`, `headers`, and `body` fields on the entry; nested `settings` / `body` objects merge, while later arrays and scalars replace earlier ones. Control per-provider request timeouts with `settings.headerTimeout` and `settings.chunkTimeout` (ms) and cap the full round-trip with `settings.timeout`; compress request bodies with `settings.encoding` set to `"zstd"`, `"gzip"`, or `"none"`. Configure Azure OpenAI via `settings.resourceName` or `AZURE_OPENAI_RESOURCE_NAME`, with Microsoft Entra ID sign-in through the Azure CLI; configure Amazon Bedrock via `settings.profile` and `settings.region`, overriding the endpoint with `settings.baseURL` for VPC or private endpoints. Sign in to OpenAI with a ChatGPT Plus or Pro plan (`Sign in with ChatGPT`, or legacy `Codex`) instead of an API key, with usage counted against the plan and no per-token cost shown; disable the built-in context-cap plugin for these sign-ins by adding `"-opencode.provider.openai-context-caps"` to the `plugins` array. The `google-vertex-anthropic` provider ID is unavailable in V2 — use `google-vertex` for Gemini, Anthropic, and OpenAI-compatible Vertex catalog models.

Add an OpenAI-compatible third-party provider so OpenCode can reach its models on day one.

json

```
// opencode.jsonc
{
  "providers": {
    "acme": {
      "name": "Acme",
      "env": ["ACME_API_KEY"],
      "package": "@opencode/ai/providers/openai-compatible",
      "settings": {
        "baseURL": "https://llm.acme.example/v1"
      },
      "models": {
        "qwen3-coder": { "name": "Qwen 3 Coder" }
      }
    }
  }
}
```

Cap request timeouts for a slow provider so hung streams fail fast and get retried.

```json
// opencode.jsonc
{
  "providers": {
    "openai": {
      "settings": {
        "headerTimeout": 30000,
        "chunkTimeout": 60000
      }
    }
  }
}
```

Attach a tenant header and extra body metadata to all OpenAI requests when routing through a gateway.

```json
// opencode.jsonc
{
  "providers": {
    "openai": {
      "headers": {
        "X-Gateway-Tenant": "engineering"
      },
      "body": {
        "metadata": { "application": "opencode" }
      }
    }
  }
}
```

Register an OpenAI-compatible third-party LLM endpoint so OpenCode can send sessions to it alongside built-in providers.

```json
// opencode.jsonc
{
  "providers": {
    "acme": {
      "name": "Acme",
      "env": ["ACME_API_KEY"],
      "package": "@opencode/ai/providers/openai-compatible",
      "settings": {
        "baseURL": "https://llm.acme.example/v1"
      },
      "models": {
        "qwen3-coder": { "name": "Qwen 3 Coder" }
      }
    }
  }
}
```

Tune per-provider timeouts for a slow private gateway and limit total request time to avoid hanging sessions.

```json
// opencode.jsonc
{
  "providers": {
    "acme": {
      "settings": {
        "headerTimeout": 30000,
        "chunkTimeout": 60000,
        "timeout": 300000
      }
    }
  }
}
```

Disable ChatGPT-session context-window caps when you want full model context regardless of plan parity with Codex defaults.

```json
// opencode.jsonc
{
  "plugins": ["-opencode.provider.openai-context-caps"]
}
```

— Every config field, env var, and package path named with runnable examples.

product docs

06 **Model variants, reasoning effort, and cache controls**

Not checked yet NEW 96

Select a model for one run without changing the default via `opencode run <provider/model>` and apply a reasoning variant with `#variant` (e.g. `opencode run openai/gpt-6.1-sol#high`). Set the default model in `provider/model` format using the `model` key, with optional `#variant` supported in agent and command references. Define or override variants in a model's `variants` array, where `settings`, `headers`, and `body` apply after provider and model values. Describe reasoning controls in `capabilities.reasoning` using the `models.dev` v2 shape (`supported`, `toggle`, `effort`, `budget`) so OpenCode builds reasoning variants automatically. Give a model a friendlier catalog ID with `modelID`, hide one from the catalog with `disabled: true`, and declare `capabilities` / `limit` to replace fallback assumptions. Set `compatibility.reasoningField` to tell OpenCode which custom field carries streamed reasoning (`reasoning`, `reasoning_content`, `reasoning_text`, or other provider-specific strings), and `compatibility.cacheFormat` to `'anthropic'` (for Claude behind OpenAI-compatible proxies like LiteLLM) or `'alibaba'`. Request a prompt-cache lifetime with `settings.cache.ttl` (a number plus unit: `ms`, `s`, `m`, `h`, `d`, e.g. `'5m'`), with model settings overriding provider settings, and keep sessions warm before cache expiry with `settings.cache.warming`; configure `settings.transport: "websocket"` for a provider. Hide free OpenCode models from a shared background server with the `OPENCODE_DISABLE_FREE_MODELS` environment variable, set via `opencode service set env OPENCODE_DISABLE_FREE_MODELS 1` or inline with `opencode serve`.

Run a single prompt with a high-reasoning variant of a model without touching your default config.

```
$ opencode run openai/gpt-6.1-sol#high "Review this migration plan"
```

Hide free OpenCode models from the catalog in a shared background server so only paid models are selectable.

```
$ opencode service set env OPENCODE_DISABLE_FREE_MODELS 1
```

Run a one-off task with a specific model and variant to get high-effort reasoning without changing your configured default.

```
$ opencode run openai/gpt-6.1-sol#high "Review this migration plan"
```

Route Claude through a LiteLLM proxy and get prompt caching with a one-hour TTL to cut repeated-context costs.

```json
{
  "providers": {
    "litellm": {
      "package": "@opencode/ai/providers/openai-compatible",
      "settings": {
        "baseURL": "https://litellm.example.com/v1"
      },
      "models": {
        "claude-opus-5-5": {
          "compatibility": {
            "cacheFormat": "anthropic"
          },
          "settings": {
            "cache": {
              "ttl": "1h"
            }
          }
        }
      }
    }
  }
}
```

Hide free OpenCode models from the catalog in a shared background server so only paid models are available to all users.

```
$ OPENCODE_DISABLE_FREE_MODELS=1 opencode serve
```

— Exact flags, config fields, and env vars named with runnable commands. product docs

07 **Plugin system: config, CLI management, and GUI extensions**

Not checked yet NEW 96

Add plugins in `opencode.jsonc` using published npm packages (`opencode-acme-plugin@1.2.0`), scoped packages (`@acme/opencode-plugin`), or local paths (`./plugins/local` , `../shared/plugin` , `/absolute/path/plugin` ,

`file:///home/me/plugins/local`); configure per-plugin options under a `package` / `options` block. Control activation by prefixing an ID with `-` to disable, `*` to match all, or `.*` for a prefix match, with a later ID re-enabling a disabled plugin; plugin arrays across `~/.config/opencode/opencode.jsonc`, `./opencode.jsonc`, and `./.opencode/opencode.jsonc` now merge from lowest to highest precedence instead of replacing one another. OpenCode auto-discovers `.ts` / `.js` files and plugin package directories under any `.opencode/plugins/` directory at project or global scope. Manage global plugins with `opencode plugin add`, `opencode plugin list` (`--builtin` for built-ins), `opencode plugin check`, `opencode plugin update`, and `opencode plugin remove`; `opencode plugin add` accepts npm names with versions/tags/ranges and Git specs (`github:`, `git+ssh://`, npm `::path:` subdirectory selectors) for branches, tags, or commit hashes. Configure TUI-only plugins separately in `cli.json` under `plugins`, where they stay active when connected to a remote server. A plugin exporting `./gui-experimental` can add panels, commands, or settings to OpenCode Desktop, configured globally and viewable under Settings > Extensions > Plugins. Two built-in plugins, `opencode.config.policy` and `opencode.provider.opencode`, ignore disable entries so a repository cannot switch off organization policy enforcement.

Pin a versioned plugin and disable all built-in provider plugins except OpenAI in a project config.

```json
// opencode.jsonc
{
  "plugins": [
    "opencode-acme-plugin@1.2.0",
    "*",
    "-opencode.provider.*",
    "opencode.provider.openai"
  ]
}
```

Install a plugin from a GitHub monorepo subdirectory using the npm : selector.

```
$ opencode plugin add
'github:acme/plugins#main::path:packages/opencode-plugin'
```

Register a TUI-only plugin that stays active when OpenCode is connected to a remote server.

```json
// cli.json
{
  "plugins": [
    "opencode-acme-cli"
  ]
}
```

Add a versioned plugin from npm and a private plugin from a Git repo to your global OpenCode config.

```
$ opencode plugin add opencode-acme-plugin@1.2.0
opencode plugin add git+ssh://git@github.com/acme/opencode-plugin.git#main
```

Load all plugins but disable every built-in provider except OpenAI, and also disable a specific reviewer plugin.

```json
{
  "plugins": [
    "*",
    "-opencode.provider.*",
    "opencode.provider.openai",
    "-acme.reviewer"
  ]
}
```

Configure a scoped plugin with specific options in your project config.

```json
{
  "plugins": [
    {
      "package": "@acme/opencode-plugin",
      "options": {
        "agent": "reviewer",
        "strict": true
      }
    }
  ]
}
```

— Config syntax and every CLI subcommand named with runnable examples. product docs

08 **Custom slash commands with templates and shell expansion**

Not checked yet NEW 96

Create custom slash commands by placing `.md` files in `.opencode/commands/` (project) or `~/.config/opencode/commands/` (global), with the trimmed Markdown body as the prompt template; nested files create path-style names (`.opencode/commands/team/review.md` becomes `/team/review`). Use `$ARGUMENTS` for the full argument string or `$1`, `$2`, ... for positional words (missing positions become empty strings; quoted groups count as single tokens). Wrap a shell command in a backtick block preceded by `!` to insert its live stdout into the prompt before submission, expanding argument placeholders first. Add YAML frontmatter with `description`, `agent`, `model` (`provider/model` or `provider/model#variant`), and `subagent`; a command-level `model` overrides the agent's configured model and the session model. Set `subagent: true` to run a command in a background child session (keeping the parent free) or `subagent: false` to force the current session. Commands can also be defined directly under a `"commands"` key in any JSON/JSONC config with `name`, `prompt`, `description`, `agent`, `model`, and `subagent` fields. Command files and config reload automatically on save.

Create a project-scoped code review command that diffs staged changes and passes them to the model automatically on each invocation.

yaml

```yaml
# .opencode/commands/review.md
---
description: Review staged changes for bugs
model: anthropic/claude-sonnet-5-5#high
---
Review this diff:
!`git diff --cached`
```

Run a background audit on a specific file without interrupting the current session by using subagent mode.

yaml

```yaml
# .opencode/commands/audit.md
---
description: Audit a file for bugs and missing tests
agent: general
subagent: true
---
Audit $ARGUMENTS for bugs and missing tests.
```

Define a command inline in the OpenCode JSON config to share it across projects without adding per-repo files.

```json
{
  "commands": [
    {
      "name": "audit",
      "prompt": "Audit $ARGUMENTS.",
      "description": "Audit a package",
      "agent": "general",
      "model": "anthropic/claude-sonnet-5-5#high",
      "subagent": true
    }
  ]
}
```

Create a project-scoped `/review` command that diffs staged changes and passes them to the model for review — no copy-paste needed.

```markdown
# .opencode/commands/review.md
Review only the staged changes: !
`git diff --cached`
```

Run a background audit in a child session so the parent session stays free while the model checks a package for bugs.

```markdown
# .opencode/commands/audit.md
---
description: Audit a package
agent: general
model: anthropic/claude-sonnet-5-5#high
subagent: true
---
Audit $ARGUMENTS for bugs and missing tests.
```

Define a command in the OpenCode config file to audit code with a fixed model and agent, available to all projects.

```json
{
  "commands": {
    "audit": {
      "prompt": "Audit $ARGUMENTS.",
      "description": "Audit a package",
      "agent": "general",
      "model": "anthropic/claude-sonnet-5-5#high",
      "subagent": true
    }
  }
}
```

— File paths, placeholders, and frontmatter fields named with runnable examples.

product docs

09 **Service debugging: CORS, heap dumps, and CPU profiling**

Not checked yet NEW 93

Check server health and PID with `opencode api get /api/info`. Configure CORS for browser clients with `opencode service set cors <origin>` and `opencode service get cors` (comma-separated HTTP/HTTPS origins), or pass `--cors <origin>` (repeatable) to `opencode serve`, which overrides the persisted list when combined with `--service`. Capture a heap snapshot of the running background server with `opencode debug heap-dump` (`--directory` for a custom output path), or target a remote server with `opencode debug heap-dump --server <url>` (using `OPENCODE_PASSWORD` if required). On macOS and Linux, send `SIGPROF` to the server PID for a ten-second CPU profile written as `cpu-<pid>-<timestamp>.cpuprofile`, or `SIGUSR1` for a heap snapshot written as `heap-<pid>-<timestamp>.heapsnapshot`, both in the log directory. Send raw HTTP requests to the local service with `opencode api`, an HTTP method and path or OpenAPI operation ID, a JSON body via `--data` / `-d`, and headers via `--header` / `-H`.

Allow a browser dev server on a local network to connect to your OpenCode service without CORS errors.

```
$ opencode service set cors "http://192.168.1.10:3001,
https://app.example.com"
opencode service get cors
```

Profile CPU usage of the background server for 10 seconds to track down a performance spike on macOS or Linux.

```
$ PID=$(opencode api get /api/info | jq -r .pid)
kill -SIGPROF $PID
```

— Every command, flag, and output filename pattern named with examples. product docs

10 **Network proxy, CA trust, and mutual TLS for providers**

Not checked yet NEW 91

Configure outbound proxying for the background service with `opencode service set env HTTP_PROXY`, `HTTPS_PROXY`, and `NO_PROXY` (default loopback exclusion `localhost,127.0.0.1,::1`), and remove a persisted variable with `opencode service unset env <VAR>`. Trust a private certificate authority by setting `NODE_EXTRA_CA_CERTS` to a PEM file path, persisted via `opencode service set env NODE_EXTRA_CA_CERTS /etc/company/ca.pem`. Present a client certificate for mutual TLS per provider with `settings.tls` (`cert`, `key`, optional `passphrase`) in `opencode.jsonc`, supporting `{file:...}` and `{env:...}` interpolation for credential values. Verify the CLI-to-service connection after changes with `opencode api get /api/info`. Since `opencode service set env` does not configure foreground processes, pass proxy and CA environment variables inline to `opencode --standalone` or `opencode serve` instead.

Route all background service traffic through a corporate proxy, excluding loopback addresses so the CLI can still reach its local service.

```
$ opencode service set env HTTP_PROXY
http://proxy.example.com:8080
opencode service set env HTTPS_PROXY
http://proxy.example.com:8080
opencode service set env NO_PROXY localhost,127.0.0.1,::1
opencode service start
```

Trust an internal CA so OpenCode can reach endpoints whose certificates are signed by a private authority.

```
$ opencode service set env NODE_EXTRA_CA_CERTS
/etc/company/ca.pem
opencode service start
```

Present a client certificate for mutual TLS when connecting to a corporate LLM gateway that requires it.

```json
{
  "providers": {
    "company": {
      "package": "@opencode/ai/providers/openai-compatible",
      "settings": {
        "baseURL": "https://llm.company.example/v1",
        "tls": {
          "cert": "{file:/etc/company/client.pem}",
          "key": "{file:/etc/company/client.key}",
          "passphrase": "{env:CLIENT_KEY_PASSPHRASE}"
        }
      }
    }
  }
}
```

— Exact env vars and config fields named with runnable examples. product docs

11 **Skills system: reusable instruction files and catalogs**

Not checked yet NEW 90

Load skill instructions and bundled resources into a conversation with the `skill` tool by supplying a skill ID, or reference one explicitly in a prompt with `@skill-id`. Skills are stored as Markdown under `.opencode/skills/<skill-id>/SKILL.md` or as root-level `*.md` files, with a path-derived case-sensitive ID; OpenCode auto-searches `~/.config/opencode/skills`, `~/.claude/skills`, `~/.agents/skills`, `.opencode/skills`, `.claude/skills`, and `.agents/skills`. Control discovery with the `opencode/autoinvoke` frontmatter key (`false` hides it from the model) or `disable-model-invocation: true`. Add local directories or HTTP catalogs as extra sources via the `skills` array in `opencode.jsonc` (relative, `~/`, absolute paths, or `http(s)://` URLs); HTTP catalogs serve an `index.json` listing skill names, file paths, and a `version` field OpenCode uses to refresh its cache. Control access per-agent or globally with `permissions` rules using `action: skill`, a `resource` skill-ID pattern, and `effect` of `allow`, `ask`, or `deny` — placed under `agents.<id>.permissions` to scope to one agent.

Create a project-level skill that walks the agent through preparing a Git release without loading it into every prompt.

```yaml
.opencode/skills/git-release/SKILL.md
```

Add a shared company skill catalog and a local team directory as additional skill sources so all engineers get the same reusable instructions.

```json
{
  "$schema": "https://opencode.ai/config.json",
  "skills": [
    "./team-skills",
    "https://example.com/opencode/skills/"
  ]
}
```

Lock down a sensitive release skill so the model must ask before loading it, while blocking all other skills from a specific agent.

```json
{
  "permissions": [
    {"action": "skill", "resource": "git-release", "effect":
"ask"},
    {"action": "skill", "resource": "*", "effect": "deny"}
  ]
}
```

Create a project-level skill that the model auto-discovers for release preparation tasks

```yaml
.opencode/skills/git-release/SKILL.md
```

Register a team HTTP catalog and an extra local directory so all project contributors share the same skill library

```json
{
  "$schema": "https://opencode.ai/config.json",
  "skills": [
    "./team-skills",
    "https://example.com/opencode/skills/"
  ]
}
```

Require the model to ask before loading the git-release skill, and block it from loading a secrets-audit skill automatically

```json
{
  "$schema": "https://opencode.ai/config.json",
  "permissions": [
    { "action": "skill", "resource": "git-release", "effect":
"ask" },
    { "action": "skill", "resource": "secrets-audit", "effect":
"deny" }
  ]
}
```

— Every config key, path, and permission field named with examples. product docs

12 **Named References for external directories and repos**

Not checked yet NEW 90

Add a local directory reference in config with `path`, an optional `description` to advertise it to agents, and optional `hidden` to remove it from interactive client selectors. Add a remote Git repository reference with `repository` (or `git`) and an optional `branch`; supported remote forms include GitHub shorthand, Git URLs, host/path forms, and SCP-style remotes, though `file:` repositories are not supported. OpenCode stores one checkout per remote and branch under `~/.local/share/opencode/repos/<host>/<repository-path>` (with an `@<branch>` suffix), cloning missing repos asynchronously and refreshing them in the background at most once every 24 hours. Access outside the active Location still follows normal tool rules and the `external_directory` permission, and an attached reference exposes only a non-recursive listing of its root's immediate files and directories.

Attach a local documentation directory as a named reference so agents can be directed to read files from it without leaving the project.

```json
// opencode config (JSONC)
{
  "references": {
    "docs": {
      "path": "~/work/product-docs",
      "description": "Use for product behavior and terminology"
    }
  }
}
```

Pin a remote SDK repo at a specific release branch so agents always read a stable version of the source.

```json
// opencode config (JSONC)
{
  "references": {
    "internal-sdk": {
      "repository": "git@gitlab.example.com:platform/sdk.git",
      "branch": "release/v2",
      "description": "Use for SDK implementation details",
      "hidden": false
    }
  }
}
```

Attach a local documentation directory so an agent can browse and read files from a shared repo outside the project.

```json
// opencode.json
{
  "references": {
    "docs": {
      "path": "~/work/product-docs",
      "description": "Use for product behavior and terminology"
    }
  }
}
```

Pin an internal SDK remote to a specific release branch so the agent always reads that version's source.

```json
// opencode.json
{
  "references": {
    "internal-sdk": {
      "git": "git@gitlab.example.com:platform/sdk.git",
      "branch": "release/v2",
      "description": "Use for SDK implementation details",
      "hidden": false
    }
  }
}
```

— Config keys, storage path, and refresh interval all named with examples. `product docs`

13 **Post-edit formatter system** Not checked yet NEW 90

Enable all built-in post-edit formatters by setting `formatters` to `true` in `opencode.jsonc`, disable globally with `false`, or clear a lower-priority config's setting with `null`. Built-in formatters auto-run (only when their executable is available) for 20+ languages including Elixir, JavaScript/TypeScript, HTML/CSS/SCSS/Sass/Less, Vue, Svelte, JSON/JSONC, YAML, TOML, XML, Markdown, GraphQL, Zig, C/C++, Kotlin, Python, R, Ruby, OCaml, Terraform, LaTeX, Bash, and Clojure. Override or define formatters with a `formatters` object where each entry supports `command` (an array, not a shell string, with `{file}` replaced by the file's absolute path), `extensions` (with leading dot), and `env` (`Record<string, string>`); disable a single built-in by setting its entry to `false`. Extension matching is case-sensitive and uses only the final extension, so `.part.md` matches `.md`, not `.part.md`.

Enable all built-in formatters so OpenCode automatically runs the appropriate formatter (e.g. a project-local Prettier) after every file edit.

json

```
{
  "formatters": true
}
```

Override the built-in Markdown formatter with a custom command and also define a fully custom formatter for a proprietary extension.

json

```
{
  "formatters": {
    "markdown": {
      "command": ["prettier", "--write", "{file}"],
      "extensions": [".md", ".mdx"]
    },
    "custom": {
      "command": ["custom-fmt", "{file}"],
      "extensions": [".custom"]
    }
  }
}
```

Disable only the built-in Deno Markdown formatter while leaving all other built-ins active.

```json
{
  "formatters": {
    "deno-markdown": false
  }
}
```

Enable all built-in formatters so OpenCode automatically runs the appropriate formatter (e.g. Prettier for JS/TS) after any file write or edit.

```json
// opencode.jsonc
{
  "formatters": true
}
```

Add a custom formatter for a bespoke file extension alongside the built-ins, injecting a required env var.

```json
// opencode.jsonc
{
  "formatters": {
    "custom": {
      "command": ["custom-fmt", "--in-place", "$FILE"],
      "extensions": [".custom"],
      "env": {
        "CUSTOM_FMT_STYLE": "compact"
      }
    }
  }
}
```

Disable the built-in markdown formatter while keeping all other built-ins active.

```json
// opencode.jsonc
{
  "formatters": {
    "markdown": {
      "disabled": true
    }
  }
}
```

— Exact config keys, defaults, and language list with runnable examples. product docs

14 **opencode run, mini, standalone, service, and paths CLI**

Not checked yet NEW 88

Run a prompt without the interactive interface using `opencode run`, for scripts and CI jobs that need model output directly in the terminal. Start a minimal interactive interface with `opencode mini` (`--help` exposes session, model, agent, prompt, and replay options). Run with a private server using `--standalone`, or connect to a specific server with `--server`; explicitly start the shared background service with `opencode service start`. Print local paths for other tools with `opencode paths`, accepting a selector of `home`, `data`, `cache`, `state`, `tmp`, `bin`, `log`, or `repos`. Pass a directory path to work in a different project without navigating to it first.

Submit a one-shot prompt from a CI pipeline without opening the interactive TUI.

```
$ opencode run "Explain this repository"
```

Pipe a local SQLite database path into sqlite3 for direct session inspection.

```
$ sqlite3 "$(opencode paths data)"
```

Run OpenCode with an isolated private server so it never touches the shared background service.

```
$ opencode --standalone
```

Submit a prompt from a CI job without opening the interactive TUI, piping model output directly to the terminal.

```
$ opencode run "Explain this repository"
```

Run OpenCode with an isolated private server so it does not share sessions or state with other local clients.

```
$ opencode --standalone
```

Print the SQLite database path for use with external tooling such as sqlite3.

```
$ sqlite3 "$(opencode path data)"
```

— Exact subcommands, flags, and selectors named with runnable examples. product docs

15 **V2 theming system with 22 built-in themes** Not checked yet NEW

87

Set the active theme and lock its mode in `~/.config/opencode/cli.json` with `theme.name` and `theme.mode` (`auto`, `dark`, or `light`). Create a custom theme by placing a JSON file in `~/.config/opencode/themes/` (global) or the project tree, where the filename (minus `.json`) becomes the theme name; open the picker from the TUI with Ctrl+P > Open settings. Built-in themes now include `aura`, `carbonfox`, `cobalt2`, `cursor`, `dracula`, `flexoki`, `catppuccin-frappe`, `github`, `material`, `mercury`, `monokai`, `nightowl`, `orng`, `lucent-orng`, `osaka-jade`, `palenight`, `rosepine`, `solarized`, `synthwave84`, `vercel`, `vesper`, and `zenburn`. V2 theme files declare a `base` token tree and at least one hue palette (modes do not inherit from the built-in theme or each other); override hue aliases (`accent`, `interactive`, `neutral`) via `$hue.<name>.<step>` references, set state colors with `$hovered`, `$focused`, `$pressed`, `$selected`, `$disabled` under action/formfield groups, and control the hues used to distinguish agents with `categorical`.

Lock OpenCode to the Dracula dark theme globally so it never follows terminal light/dark switching.

```json
{
  "theme": {
    "name": "dracula",
    "mode": "dark"
  }
}
```

— Config keys and full theme list named with one runnable example. product docs

16 **Code Mode execution namespace: execute, browser, and opencode tools**

Not checked yet NEW 85

The `execute` tool runs JavaScript to call and combine catalog tools in parallel, process results without flooding model context, and access `fetch` for HTTP; it can be removed entirely by denying resource `*`. The `browser` namespace controls an attached browser with tab and navigation commands, page snapshots, clicking and form input, screenshots, console and network inspection, performance traces, heap inspection, and Lighthouse audits, with file transfers capped at 5 MiB each. The `opencode` namespace adds `worktree_create` (returns `directory`), `worktree_list`, `worktree_remove` (accepts `force` to discard uncommitted changes), `session_rename` (accepts optional `sessionID`), and `session_move` (accepts relative or absolute `directory`).

Run two file reads in parallel inside Code Mode to cut round-trip time when gathering context from multiple files.

> 📍 Read package.json and README.md in parallel using execute, then return their relevant setup details.

Open a live page in the attached browser, snapshot its content, and extract information — useful for verifying a local dev server or scraping a login flow.

> 📍 Open `https://example.com`, take a snapshot of that tab, and report the main heading.

— Full tool catalog named with exact parameters and limits. product docs

17 **Compaction and experimental session controls** Not checked yet

NEW 82

Opt into native provider-side compaction instead of local summaries with `settings.summary: "native"` (shown in examples as `"compaction": "native"`) on a provider or model; control the retained tail and disable new automatic compaction with `compaction.keep.tokens`, `compaction.keep.buffer`, and `compaction.auto: false`; and avoid pausing the session by writing the summary early with `experimental.background_compaction: true`. Related experimental flags add a `todowrite` tool for tracking multi-step work via `experimental.todo`, cap subagent recursion with `experimental.subagent_depth`, and control how long an idle project location stays loaded with `experimental.location_idle_timeout` (accepting durations like `"2 hours"` or `"never"`).

Opt into native provider-side compaction for a specific model so the provider handles context summarisation instead of the local engine.

```json
{
  "providers": {
    "openai": {
      "models": {
        "gpt-6.1-sol": {
          "settings": {
            "compaction": "native"
          }
        }
      }
    }
  }
}
```

Enable background compaction and the todowrite tool for long multi-step agentic sessions so compaction never blocks the conversation.

```json
{
  "experimental": {
    "background_compaction": true,
    "todo": true
  }
}
```

— Config keys and defaults named, with one runnable example. product docs

18 **File editing tools: patch, edit, write** Not checked yet NEW

78

`patch` applies multi-file changes using `*** Add File`, `*** Update File`, and `*** Delete File` sections (available for supported GPT models); `edit` replaces targeted text via `path`, `oldString`, and `newString`, setting `replaceAll` to true when the old text appears more than once; `write` creates or fully replaces a text file via `path` and `content`, automatically creating missing parent directories and optionally running the configured formatter.

— Exact parameters and formats named, no runnable command shown. product docs

19 **File search and read tools: grep, glob, read** Not checked yet

NEW

75

`grep` searches file contents via a regex `pattern` (or `literal` set to true), narrowed by `path`, `include`, `caseSensitive`, and `limit`, defaulting to 100 matching lines

with a 30-second timeout. `glob` finds file paths by pattern via `path`, `hidden`, and `limit`, also defaulting to 100 files with a 30-second timeout. `read` reads files or non-recursive directory listings via `path`, `offset`, and `limit`, capping text reads at 2,000 lines and 50 KiB per page while passing PNG, JPEG, GIF, WebP, and PDF files up to 20 MiB to the model.

— Exact defaults and limits named, no runnable example. product docs

20 **Subagent orchestration tool** Not checked yet NEW 73

Start child sessions with the `subagent` tool by supplying an agent ID, `description`, and `prompt`; the call returns immediately with a `sessionID` and the parent receives `<subagent>` messages as the child works, with default nesting depth of one. Cap recursion depth with `experimental.subagent_depth` in config.

Delegate an open-ended exploration task to a child session and get a structured result back without blocking the parent.

> 📍 Ask the explore subagent to map the authentication flow and return the key files.

— Return behaviour and config cap named, one example shown. product docs

21 **Image generation tool** Not checked yet NEW 70

Generate or edit images with the `experimental_image_generate` tool by supplying a `prompt` and optionally `images`, `model`, or `path`; configure its default model via `experimental.image_generate` (as `providerID/modelID`, optionally with `#variant`) in config.

— Named tool and config key, but no example given. product docs

22 **Core config keys: shell, update, worktree directory, snapshots, output limits**

Not checked yet NEW 68

Configure the shell used by terminal and shell tools with the `shell` key, and control update behaviour per machine with `update` set to `"disable"`, `"prompt"` / `"manual"`, or `"auto"` (default; project-level values are ignored). Set the parent directory for new local worktrees with `worktree.directory`, resolved relative to the project's canonical checkout. Enable or disable filesystem snapshots for undo/revert with the `snapshots` key. Control how oversized images loaded by the `read` tool are resized or rejected with the `media` key. Cap tool output size with `tool_output.max_lines` and `tool_output.max_bytes`. Reference environment variables and file contents inside config strings using variable substitution in opencode.json(c).

Pin worktree output to a per-project subfolder so new lanes don't scatter across the filesystem.

```json
{
  "worktree": {
    "directory": ".lane/trees"
  }
}
```

— Named keys listed thinly, mostly without worked examples. product docs

---

23 **Web access tools: fetch and web_search** Not checked yet NEW

65

`fetch` retrieves HTTP or HTTPS URLs as `markdown`, `text`, or `html` (defaulting to `markdown`, with `timeout` accepting up to 120 seconds); `web_search` searches a connected web integration for current information via a `query`, with providers, rate limits, and disabling documented separately.

— Formats and timeout named; no example given. product docs

THINNER COVERAGE BELOW

24 **Shell execution tool** Not checked yet NEW

55

Execute shell commands via the `shell` tool with `workdir`, `timeout` in milliseconds (default two-minute foreground), and `background` for long-running processes like dev servers.

— Parameters named but no example shown. product docs

---

25 **Interactive multi-choice ask tool** Not checked yet NEW

50

Present interactive multi-choice questions to the user mid-task with the `ask` tool; each question carries a header, prompt, and choices, with `multiple` allowing more than one selection.

— Describes fields but no usage example. product docs

└──▷ ALSO FROM THESE RELEASES

Keep prompt-cache hits alive and warm just before each model's cache expires, up to 30 minutes, to avoid cold-start latency on long sessions.

```json
{
  "providers": {
    "anthropic": {
      "settings": {
        "cache": {
          "warming": "auto",
          "ttl": "5 minutes"
        }
      }
    }
  }
}
```

└──▷ BREAKING ON UPGRADE

- ! V1 frontmatter fields in agent Markdown files are deprecated; files using them are still read but new config should use V2 fields. The top-level `maxSteps` field is explicitly called out: 'Do not use legacy top-level fields such as `maxSteps` in new V2 agent configuration.'
- ! Before OpenCode 2.2.3, `integration.use` covered MCP servers, skills, and plugins through an `mcp:`, `skill:`, or `plugin:` prefix; a broad `integration.use` deny such as `*` no longer blocks MCP servers, skills, or plugins — replace it with separate `mcp.use`, `skill.use`, and `plugin.use` deny statements.
- ! Clients older than 2.2.3 ignore `integration.connect`, `integration.use` (in its new meaning), `mcp.use`, `skill.use`, and `plugin.use` — require client 2.2.3 or newer wherever you rely on those policy actions.
- ! Clients older than 2.2.6 ignore `websearch.use` — require client 2.2.6 or newer wherever you rely on web search policy control.
- ! The `google-vertex-anthropic` provider ID is unavailable in V2; use the `google-vertex` provider ID instead for Gemini, Anthropic, and OpenAI-compatible Vertex catalog models.
- ! Before OpenCode 2.2.3, `integration.use` covered MCP servers, skills, and plugins via `mcp:`, `skill:`, or `plugin:` prefixes; unprefixed `integration.use` statements (such as a bare `*` deny) now apply to integrations only and no longer block MCP servers, skills, or plugins — replace them with equivalent `mcp.use`, `skill.use`, and `plugin.use` statements.
- ! Clients older than 2.2.3 ignore `integration.connect`, `integration.use` (in its new meaning), `mcp.use`, `skill.use`, and `plugin.use` — upgrade to 2.2.3 or newer wherever you rely on these policy actions.
- ! Clients older than 2.2.6 ignore `websearch.use` — upgrade to 2.2.6 or newer to enforce web search provider restrictions.
- ! V1 frontmatter fields are deprecated; files using them are still read under compatibility rules but `maxSteps` and other legacy top-level fields must not be used in new V2 agent configuration.

! `autoapprove` is deprecated; use top-level `policies` instead. The old key still works and merges into `policies`.

updated · Nov 24, 2026 NOTES ↗ LATEST STABLE

OpenCode adds browser import, automation scheduling via `/automation`, video generation, and a new TUI look in this snapshot release

└──▷ **USE IT**

Disable the new automations subsystem entirely if you do not want the agent running scheduled tasks

```json
{ "automations": false }
```

Adds automation support: schedule agent prompts, scripts, or shell commands on a timer, cron schedule, or MCP server event, managed with `/automation` (disable via `"automations": false` in your config), and lets plugins automate workflows programmatically using `ctx.experimental.automation.create`, `ctx.experimental.automation.list`, and `ctx.experimental.automation.delete`. Not checked yet

Import sign-ins and browsing history from Chrome, Edge, Firefox, or Safari into the built-in desktop browser, and add extensions from Chromium browsers or the Chrome and Edge stores Not checked yet

Generate videos with `generate_video` when the connected provider has a video model; files are saved into your workspace when rendering completes Not checked yet

Config-only providers such as Ollama now show their models in `/models` without requiring a `/connect` login Not checked yet

New TUI designs are on by default with the `opencode` theme; switch to `opencode-classic` in Theme settings to restore the previous colors Not checked yet

+3 more

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.4755 CODE ↗ RANK

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

> Magpie's artifact runtime moved closer to full claude.ai compatibility this window, gaining a per-artifact SQLite document store with the `db` / `user` / `permissions` capabilities and a self-publishing API via claude.use("artifact"), alongside TOML config support for agent plugins, per-plugin local PII masking, automatic adoption of manually copied skill folders, and a more informative Artifacts empty state.

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

01 **Per-artifact db store and claude.ai runtime compatibility**

Not checked yet NEW 88

Adds a per-artifact SQLite document store at `data.db` (beside an artifact's versions folder) implementing claude.ai's `db` capability, with path validation up to `maxSegments` 16 and `maxPathBytes` 1000, a document size cap `maxDocBytes` of 256 KB, store-wide caps `maxDocs` 25,000 and `maxDataBytes` 256 MB, and lease durations clamped via `minLease`, `maxLease`, and `defaultLease` 30s between 1s and 10min. Also introduces window.claude.use(name) as a client-side runtime injected into every artifact page's `<head>`, exposing the `db`, `user`, and `permissions` capabilities from claude.ai's artifact contract 0.2.99 so pages written for claude.ai run unmodified, with agents reaching an artifact's full data store via the Go API OpenDB(artifactID, version, Viewer) and closeDB(artifactID) through `artifact_data` while a page viewer sees only its own scoped data.

– Full limits and API names given, but Go API is dev-facing not end-user runnable

v0.1.4754

02 **TOML config support for agent plugins** Not checked yet NEW 84

Agent plugins now support TOML configuration files alongside existing JSON and YAML, enabling integration with agents whose configs are TOML (such as Codex), writing dotted key paths like `model_providers.magpie.base_url` as TOML tables that preserve existing keys and comments.

Connect a TOML-configured agent (such as Codex) by writing individual dotted key paths into the agent's TOML config file, preserving all existing keys and comments.

```shell
export const agent = { id: "harness", name: "Harness", config:
"~/.harness/config.toml", model: "model" };
export function connect({ gateway, model }) {
  return {
    model_provider: "magpie",
    "model_providers.magpie.base_url": gateway.v1,
    "model_providers.magpie.bearer_token": gateway.key,
    "model_providers.magpie.wire_api": "responses",
  };
}
```

— Includes a runnable config connector example with exact keys v0.1.4751

03 **Artifact self-publishing via claude.use** Not checked yet NEW

83

A page can activate self-publishing by calling claude.use("artifact"), letting it push new versions of itself — its whole HTML or just changed files — through the `<page>/_api/` endpoint. The runtime config now exposes `artifact` and `self` capability flags so pages can check availability via `served.artifact` and `served.self`.

— Exact call and endpoint named, no worked example given v0.1.4755

04 **Per-plugin local PII masking models** Not checked yet NEW

65

Select per-plugin local PII masking models in Settings › Privacy, configured with `id`, `name`, `path`, `minScore`, and `revision` fields under a `models` key in the plugin's options.

— Exact config fields and UI path given, no worked example v0.1.4753

05 **Automatic adoption of manually copied skill folders**

Not checked yet IMPROVED

60

Skill folders placed manually into the library's skills folder are now automatically listed and adopted on the next full page read, excluding dot-folders, folders missing a `SKILL.md`, magpie's own leftover copies, and previously removed skills.

— Clear behaviour and exclusions but no command or flag to invoke v0.1.4748

THINNER COVERAGE BELOW

06 **Informative empty state for Artifacts page** Not checked yet

IMPROVED

55

The Artifacts page empty state now shows a description of what an artifact is, copyable example prompts (a git-commit report with a chart, an architecture diagram, a Pomodoro timer), and which agents have access to the Artifacts server — or a prompt to enable it in the Library.

— Describes UI content but gives no config surface or command `v0.1.4747`

WAS THIS USEFUL? 👍 👎

OpenQodex

Feedback ⓘ openqodex@0.23.0 NOTES ↗ RANK

OpenQodex is a join the world's most widely adopted, AI-powered developer platform where millions of developers, businesses, and the largest open source community build software that advances humanity.

> OpenQodex added persistent and per-run controls for the Codex reviewer's model and reasoning effort, and substantially expanded call-graph accuracy — surfacing `unknown.gaps` across callers, callees and impact answers, flooring callers counts for unparsed or dynamic route code in Next.js, Nest, FastAPI, Go and Express, improving router prefix resolution, and fixing silent golangci-lint failures.

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

01 Per-review Codex reviewer model and effort controls

Not checked yet NEW 95

Config keys `reviewer_model` (a model ID) and `reviewer_effort` (`low`, `medium`, `high`, or `xhigh`) in `~/.openqodex/config.yaml` set persistent defaults for the Codex reviewer; `--reviewer-model` and `--reviewer-effort` flags on the `review` command override them for a single run. `openqodex doctor` now names the active reviewer model and effort level in its output.

Pin the Codex reviewer to a specific model and cap its reasoning at medium effort for all reviews on this machine.

yaml

```yaml
reviewer_model: o3
reviewer_effort: medium
```

Override the reviewer model and effort for a single review run without touching the config file.

```
$ openqodex review --reviewer-model o3 --reviewer-effort high
```

— Exact config keys, flags and runnable commands given openqodex@0.23.0

02 Callers floor for unreadable or dynamic route code

Not checked yet IMPROVED 80

OpenQodex now reports a `callers` floor, naming the file, when a Next.js route handler or Nest controller action lives in a file over 256 KB that a framework plugin did not read.

It also floors `callers` and names the cause when a FastAPI route uses an f-string path, a Go route uses a constant longer than the string limit, or an Express route uses a computed path — previously those handlers were silently dropped. The floor now also applies to FastAPI dependencies written past reader limits (long dotted names, deep parentheses, long literal paths, parameters past the 64th), Nest controllers inside functions late in long files, guards past the 64th in one `UseGuards`, and routes inherited by classes with over 1,000 members.

— Enumerates every triggering pattern and threshold `openqodex@0.22.2`

---

03 **Router mount prefix resolution for dynamic and re-exported routers**

Not checked yet IMPROVED 80

Express routers mounted under a computed prefix — app.use(process.env.PREFIX, router) or `app.use(` /${version}`, router)` `— now report their routes as unknown with a computed path rather than listing them without the prefix; middleware under such a prefix no longer appears to apply to routes outside it. Routers exported through` export { router }`,` export { router as api }`, or` export default router `are now mounted under their prefix (e.g.` `/v2` /things `instead of` `/things`); Express, Hono, and FastAPI no longer mount a whole router when only a member such as` router.handler is passed.

— Names exact code patterns and resulting path behavior `openqodex@0.22.2`

---

04 **Graph gap reporting via `unknown.gaps` field** Not checked yet

NEW 75

A new `unknown.gaps` field on `callers`, `callees`, and `impact` answers names each place the call graph gave up — computed loads, unresolved calls, unknown-type member reads, unread files, cut walks — with file, line, and what to check next. `callers` answers more than one hop deep, and `impact` answers at their second hop, now say may be incomplete and name the caller when any caller walked through has a graph or framework gap.

— Names the field and gap types, no query example given `openqodex@0.22.2`

---

05 **Call graph recognizes decorator and reflection-based calls**

Not checked yet IMPROVED 68

`graph callers` now lists places a Python function is used as a bare decorator — `@wrap` or `@pkg.wrap`, including on methods — not only @wrap() calls. Reflection-based call sites — Java `getMethod` and `getDeclaredMethod`, Go `MethodByName`, and Python `getattr` — are now recorded as gaps in the call graph, flagging that a method found by a runtime-computed name may be a hidden caller.

— Names exact syntax patterns across four languages `openqodex@0.22.2`

THINNER COVERAGE BELOW

06 **golangci-lint failure reporting for unchecked Go modules**

Not checked yet IMPROVED 55

OpenQodex now reports `golangci-lint` as failed when one Go module in the change cannot type-check, even when another module produced findings, so unchecked packages are no longer silently hidden.

— Describes behavior but no flag or config to control it `openqodex@0.22.2`

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.173.0-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 shipped streaming and batched app-server file APIs, a per-tool approval reviewer in `config.toml`, a configurable custom status line, and a wave of terminal UX improvements including slash-command Tab completion, thread-link navigation, and Orca terminal recognition.

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

04 **Slash-command Tab completion** Not checked yet NEW 70

The TUI now completes arguments for supported slash commands when pressing Tab — for example, typing `/mcp lo` completes to `/mcp login`.

— Gives an exact worked example of the feature rust-v0.172.0

05 **Per-tool approval reviewer in config.toml** Not checked yet NEW 65

Administrators can now set an approval reviewer for individual connected app tools in `config.toml`.

— Names the config file but not key syntax rust-v0.172.0

06 **Task-group keyboard shortcuts** Not checked yet NEW 60

Jump between task groups in the agent command center using Shift+Up and Shift+Down.

— Exact key combination a reader can use immediately rust-v0.172.0

THINNER COVERAGE BELOW

07 **Code Mode now uses gRPC unconditionally for host connections**

Not checked yet 55

BREAKING

The `code_mode_host_grpc` setting is deprecated — Code Mode now uses gRPC for all host connections unconditionally.

— Names the deprecated setting and new unconditional behavior rust-v0.172.0

08 **Thread-link navigation** Not checked yet NEW 50

Users can follow thread links from search results and transcripts to open saved conversations and nested agents directly in the terminal.

— Describes behavior but no exact UI path or command `rust-v0.172.0`

09 **Amazon Bedrock Runtime selection during setup** Not checked yet

NEW

50

App-server clients can now select Amazon Bedrock Runtime during setup or sign-in.

— Names the integration but no setup steps `rust-v0.172.0`

12 **MCP server connection improvements** Not checked yet IMPROVED

45

Optional stdio MCP servers with cached tool lists now start on their first tool call rather than at launch, MCP connection sharing is now enabled by default so multiple agent turns reuse existing MCP connections rather than opening new ones, and OAuth discovery is now enabled by default for config servers.

— Explains before/after behavior but no config surface Explains the mechanism but gives no config override One-line description with no further mechanism

`rust-v0.172.0` `rust-v0.173.0-alpha.5`

└──▷ **BREAKING ON UPGRADE**

! The `code_mode_host_grpc` setting is deprecated — Code Mode now uses gRPC for all host connections unconditionally.

WAS THIS USEFUL?  

Factory Feedback ⓘ

CLI v1.19.0, Desktop v1.19.0: MCP sign-in consent option, all Jira projects, and clearer

BLOG ↗ RANK

Factory's Droids are AI software-development agents with a CLI, web app and API.

> Factory's latest CLI and Desktop releases add finer-grained MCP server controls (per-group scoping and a sign-in consent bypass), new Bugsnag and enhanced Glean connectors, quick actions in the Skills browser, shareable session draft links, in-app find-in-page search, clearer connector call previews, and a leaner organization users API.

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

01 **On-demand fields in organization users API** Not checked yet

IMPROVED 65

The organization users API now returns `role`, `status`, and `groups` fields only when explicitly requested, reducing response payload size by default.

– Names exact fields but not endpoint path or method.

CLI v1.19.0, Desktop v1.19.0: MCP sign-…

02 **Find in page in desktop app** Not checked yet NEW 65

Search within a page in the desktop app using Cmd+F to open a find bar that highlights and steps through matches.

– Exact shortcut given, immediately usable. CLI v1.14.0, Desktop v1.14.0: Find in p…

03 **MCP servers scoped to user groups** Not checked yet NEW 60

Admins can now assign MCP servers to a specific user group on top of the organization-wide allowlist, via group settings.

– Clear UI starting point, no exact config surface named.

CLI v1.17.0, Desktop v1.17.0: Skills br…

04 **Bugsnag and Glean connector updates** Not checked yet NEW 60

Factory adds a Bugsnag connector listed under the Observability category in Connector settings, with its own logo and description. Glean connectors now display Glean's logo, name, and description, and their read-only tools run without requiring approval.

– Names both connectors and their category/UI location.

CLI v1.17.0, Desktop v1.17.0: Skills br…

THINNER COVERAGE BELOW

05 **Skills browser quick actions** Not checked yet NEW 55

The Skills browser now adds Use in chat , Create skill , and Open file actions for local skills.

– Exact action labels given, clear UI entry point.

CLI v1.17.0, Desktop v1.17.0: Skills br…

06 **MCP sign-in consent bypass option** Not checked yet NEW 50

Admins can set an MCP server to bypass the forced sign-in consent step, unblocking automated sign-in flows.

– Names the setting's effect but not the exact config key.

CLI v1.19.0, Desktop v1.19.0: MCP sign-…

07 **Session draft links** Not checked yet NEW 45

Opening a New Session link with pre-loaded text now lands recipients directly in a ready-to-send draft composer instead of a blank session.

– Describes behavior but no link format or API detail.

CLI v1.18.0, Desktop v1.18.0: New sessi…

08 **Connector call previews** Not checked yet IMPROVED 45

Connector call previews now show what a pending call would do instead of raw JSON while awaiting approval.

– Describes the change but no UI path or format detail.

CLI v1.14.0, Desktop v1.14.0: Find in p…

WAS THIS USEFUL? 👍 👎

mvschwarz OpenRig

Feedback ⓘ v0.7.22 NOTES ↗ RANK

OpenRig is a network manager for AI coding agents: build persistent teams with roles, shared context and owned work.

> OpenRig's v0.7.22 release adds MCP person-token association, parallel seat restores, and tighter guards around queue updates and daemon identity polling, rounding out CLI reliability work with timeout controls and clearer failure codes.

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

01 **MCP token-to-person association and ESM import** Not checked yet

NEW 90

Associate a token file with a person so it can answer that person's questions over MCP using `rig mcp token add … --person <entity_id>`, and import the MCP module from an ESM host via `@openrig/cli/mcp.js`.

— Full command and import path given, mechanism explained v0.7.22

02 **Queue update state guard** Not checked yet NEW 80

`rig queue update --expect-state` guards queue updates against unexpected state transitions in concurrent workflows; older daemons ignore the flag.

Conditionally update a queue row only when it is still in the expected state, preventing accidental overwrites in concurrent workflows.

```
$ rig queue update <row-id> --expect-state waiting
```

— Flag, mechanism, and compatibility note all given v0.7.22

03 **Parallel seat restores via --concurrency** Not checked yet NEW 75

Run multiple seat restores in parallel to speed up recovery of a large rig after a daemon restart using `rig start --concurrency 4`.

Run multiple seat restores in parallel to speed up recovery of a large rig after a daemon restart.

```
$ rig start --concurrency 4
```

— Runnable example given but benefit left generic v0.7.22

04 **rig up timeout and failed-seat naming** Not checked yet IMPROVED

70

`rig up --timeout <seconds>` sets the CLI's wait time for a rig to come up, and `rig up` now names seats that failed to launch.

— Exact flag and behaviour change named v0.7.22

05 **Blocking poll for queue task answers** Not checked yet IMPROVED

60

Poll `rig_queue_show` with an optional `waitSeconds` argument so an agent can block until its task has an answer.

— Argument named but no usage example shown v0.7.22

06 **Daemon bind failure exit code** Not checked yet IMPROVED

60

A daemon that cannot bind now exits with a single `OPENRIG_LISTEN_FAILED` line at exit code 69.

— Exact error string and exit code given v0.7.22

THINNER COVERAGE BELOW

07 **Identity poll interval config key** Not checked yet NEW

55

Tune the cadence of identity checks with the `daemon.identity_poll_interval_ms` config key.

— Config key named but no default or range given v0.7.22

08 **Intake row naming in stream show** Not checked yet IMPROVED

40

`rig stream show` now names an item's intake row for easier trace-back.

— Brief description, no mechanism or example v0.7.22

WAS THIS USEFUL?  

Anthropic Claude Code

Feedback ⓘ v2.1.331 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.331 expands enterprise gateway authentication and proxy handling, extends Code Mode to four more cloud platforms, and adds several VSCode editor integrations plus new diagnostics and observability fields.

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

01 Gateway OIDC configuration for groups and proxy sign-in

Not checked yet NEW 88

Adds `oidc.groups_jsonpath` to read groups from matching entries in a list of objects using a JSONPath subset (RFC 9535), and `CLAUDE_GATEWAY_IDP_CONNECT_BY_HOSTNAME=1` to pass an `https` IdP hostname to a proxy during sign-in even when `NO_PROXY` is set, when `oidc.use_proxy: true` is configured.

Allow a gateway proxy to reach an OIDC IdP by hostname even when `NO_PROXY` would otherwise exclude it.

```
$ export CLAUDE_GATEWAY_IDP_CONNECT_BY_HOSTNAME=1
claude
```

— Names exact config keys and env var with a runnable example v2.1.331

02 API connectivity check in claude doctor

Not checked yet IMPROVED 75

`claude doctor` now includes an API connectivity section that reports whether the API can be reached, the HTTP status or error, or explains why connectivity was not tested.

Troubleshoot API reachability problems — including HTTP status codes or proxy errors — without leaving the terminal.

```
$ claude doctor
```

— Runnable command with described diagnostic output v2.1.331

03 **VSCode extension chat and merge actions** Not checked yet NEW

75

Adds `Add Cell to Chat` to the notebook cell menu (attaches a cell with its outputs or error to the chat), `Fix with Claude Code` as a quick fix for errors, warnings, and failed tests in the test peek and Test Results view, `Add to Chat` to the right-click menu of changed files in the Source Control view, and `Resolve Merge Conflicts` to the editor title of a file with merge conflicts and to the Merge Changes group in Source Control.

— Names exact UI locations for each new action v2.1.331

---

04 **Disable response compression behind truncating proxies**

Not checked yet NEW

70

Set `CLAUDE_CODE_DISABLE_RESPONSE_COMPRESSION` to request uncompressed model responses when behind a proxy that truncates compressed replies.

Unblock Claude Code behind a corporate proxy that silently drops compressed HTTP responses.

```
$ export CLAUDE_CODE_DISABLE_RESPONSE_COMPRESSION=1
claude
```

— Single env var with runnable shell example v2.1.331

---

05 **Optional plugin dependencies in manifests** Not checked yet NEW

70

Declare `optionalDependencies` in plugin manifests to list plugins that install alongside the declaring plugin but can be individually disabled or uninstalled.

— Names manifest field but no example given v2.1.331

---

06 **gen_ai trace span fields for LLM requests** Not checked yet

IMPROVED

65

New `gen_ai.usage.*`, `gen_ai.response.model` and `gen_ai.operation.name` fields are now added to `claude_code.llm_request` trace spans; `gen_ai.usage.input_tokens` includes prompt cache tokens.

— Named fields but no direct user action described v2.1.331

THINNER COVERAGE BELOW

---

07 **Code Mode on four new cloud platforms** Not checked yet NEW

50

Code Mode now supported in terminal sessions on Amazon Bedrock, Google Cloud's Agent Platform, Microsoft Foundry and Claude Platform on AWS.

— Lists platforms but no setup detail v2.1.331

WAS THIS USEFUL? 👍 👎

Red Hat ripwire

Feedback ⓘ v0.7.0 NOTES ↗

With ripwire, coding agents get a ranked repository map showing call graphs, blast radius and test coverage.

ripwire v0.7.0 adds verb routing, a tangle quality kind, error-masking detections, Bash `--flags`, Proto indexing, a memory ceiling, warm-run caching, and answer-size caps.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.7.0 https://github.com/redhat-et/ripwire.git
# already have the repo? check out this version:
$ git checkout v0.7.0
```

Route a `--for` question that names its symbol in backticks and asks who calls it, where it is used, or what breaks to `--callers`, `--uses`, or `--impact` automatically; opt out with `--no-verb-route` or MCP `no_verb_route`. Not checked yet

Add a `tangle` quality kind to `--quality-delta` — the thirteenth kind — detecting long functions with deep nesting, growing deep-region share, or multiplying steering locals, with a best-since-tag ratchet and an `<improved>` section. Not checked yet

Detect error-masking with new `--quality-delta` facets: `facet="failure-default"` for dropped I/O or child status, `defect="unchecked-status"` for unchecked status, `defect="unhandled-arm"` for a classifier that leaves a value without an outcome, and `similar=` for adjacent same-typed parameters (report-only). Not checked yet

Run `--quality-delta` over a pull request's whole range with the `qd-range` CI job and `scripts/qd_range.sh`. Not checked yet

Apply an allocator-level memory ceiling on by default, so a runaway step now exits with code 5 instead of taking the host down. Not checked yet

+7 more

NEW SUPPORTED PLATFORM Protocol Buffers (.proto) indexing

NEW SUPPORTED PLATFORM Bash flag analysis via --flags

UPDATED SUPPORTED PLATFORM C#, C++ and Java base and import clause indexing via shared tags-capture paths

UPDATED SUPPORTED PLATFORM Rust call-graph and macro resolution improvements

UPDATED SUPPORTED PLATFORM C and C++ smart-pointer and namespace call resolution improvements

UPDATED SUPPORTED PLATFORM Go method expression references

UPDATED SUPPORTED PLATFORM Python import and function-level module call resolution

UPDATED SUPPORTED PLATFORM TypeScript paths-alias call binding

UPDATED SUPPORTED PLATFORM Clojure metadata call extraction

UPDATED SUPPORTED PLATFORM Elixir keyword tables moved to tags.scm with string-predicate evaluation

---

WAS THIS USEFUL?  

All Hands AI OpenHands

Feedback ⓘ v1.44.0 NOTES ↗

RANK

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

> OpenHands v1.44.0 adds conversation mentions, automation visibility controls, an in-UI key-value store editor, and a new Factories sidebar.

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

01 **Conversation mentions in chat** Not checked yet NEW 50

Conversation mentions can be enabled in chat using the `conversation_mentions` feature flag.

– Names exact flag but not behavior once enabled v1.44.0

02 **Key-value store editor for automations** Not checked yet NEW 40

Users can view and edit an automation's key-value store directly in the UI.

– Clear UI capability but no navigation path or schema detail v1.44.0

03 **Factories sidebar** Not checked yet NEW 40

A new Factories sidebar adds list and drill-in views for managing factories.

– Describes views but not what a factory is or does v1.44.0

04 **Automation visibility setting** Not checked yet NEW 30

A new automation visibility setting controls who can see a given automation.

– Describes capability but no config name or options v1.44.0

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.118 NOTES ↗

RANK

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

> Graphify adds project-level control over documentation demotion via `.graphifyrc` and bumps its AST cache schema, triggering automatic re-extraction for several compiled languages.

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

01 **AST cache schema bump to v53** Not checked yet BREAKING

80

The AST cache schema version bumps from 51 to 53; the first build after upgrading automatically re-extracts C#, C, and Objective-C files, which alters node/edge output for those languages. Use `graphify extract . --force` to repair a cached graph (e.g. one that lost Svelte barrel imports) so cross-file call edges resolve correctly.

Repair a cached graph that lost Svelte barrel imports after upgrading, so cross-file call edges resolve correctly.

```
$ graphify extract . --force
```

— Names schema versions, affected languages, and repair command v0.9.118

02 **Doc demotion settings in `.graphifyrc`** Not checked yet NEW

75

Projects can now set doc-demotion behaviour via `demote_docs` or `query_demote_docs` keys in `.graphifyrc` instead of relying only on the CLI flag; an explicit CLI flag still overrides the file setting.

— Names exact config keys and override behaviour v0.9.118

└──▷ **BREAKING ON UPGRADE**

! The AST cache schema version bumps from 51 to 53; the first build after upgrading re-extracts C#, C, and Objective-C files automatically, which alters node/edge output for those languages.

WAS THIS USEFUL? 👍 👎

Yuval Sarel cones

Feedback ⓘ v0.4.31 NOTES ↗ RANK

Cones is a terminal dashboard for managing coding-agent sessions across projects.

> Cones fixed Codex session recovery from stale daemon pid files and added context-usage and plan columns to the session list, while also improving tag deletion, state-file persistence, and the background meaning index's memory footprint.

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

01 **Context-usage and plan columns in session list** Not checked yet

NEW 67

The session list now shows a context column that meters reported window usage — grey below half, amber from half, red at 80% — and falls back to a bare token count when no window is reported. It also shows a plan column with the name the harness reports (`Ultra` , `supergrok` , etc.) instead of a generic label like `max` .

— Describes thresholds and values but only a UI path to act on v0.4.29

02 **Codex session recovery from stale pid files** Not checked yet

IMPROVED 63

Codex sessions started with a stale daemon pid file now display their model, cost, plan and context, and `cones launch` , `cones wait` and the parent link correctly find their thread.

— Names affected commands but no reproduction steps given v0.4.31

03 **Lower-priority background meaning index with reduced memory**

Not checked yet IMPROVED 60

The background meaning index now runs at background priority and embeds eight passages per pass, reducing peak memory from about 1.6 GB to about 0.65 GB.

— Concrete numbers and mechanism but nothing for reader to do v0.4.29

THINNER COVERAGE BELOW

04 **Tag deletion sweeps all sessions** Not checked yet IMPROVED 38

Delete in the places checklist now removes a tag from every session that carries it, not just the current one.

— Clear behavior change but thin on mechanism v0.4.29

05 **Batched state file writes** Not checked yet IMPROVED 35

The dashboard now writes its state files in one call instead of thousands of small writes, keeping the UI responsive under load.

— Internal optimization with no user-facing action v0.4.29

WAS THIS USEFUL? 👍 👎

Letta Code Feedback ⓘ v0.34.70 NOTES ↗ RANK

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

> Letta Code's single release in this window adds task-description scoping for subagents, giving delegated work a more focused context.

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

01 **Task-scoped subagents** Not checked yet NEW 31

Subagents can now be scoped to a specific task description, giving each subagent a focused context for the work it is delegated.

— Describes the behavior but no mechanism, flag, or usage detail. v0.34.70

WAS THIS USEFUL? 👍 👎

jcode

Feedback ⓘ v1.1.2 NOTES ↗ RANK

Jcode is a high performance coding agent harness written in Rust, built for heavy multi-agent dev workflows

> jcode's v1.1.2 release made small refinements to the live status line and model naming display.

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

01 **Streaming and finalizing labels in status line** Not checked yet

IMPROVED 30

The live status line now shows streaming and finalizing activity labels, with yellow highlights used to indicate stalled streams.

— Describes what changed but no mechanism or location detail v1.1.2

02 **Cleaner effort-named Gemini model labels** Not checked yet IMPROVED

25

jcode now omits the redundant effort suffix on effort-named Gemini models.

— Names the affected models but no further detail v1.1.2

WAS THIS USEFUL? 👍 👎

Cotool

Feedback ⓘ v0.81.0 NOTES ↗ RANK

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

> Cotool's November 23 release expanded its integration ecosystem with nine new integrations and five managed log ingestion sources, added new AI models including GPT-6.2 and Gemini 4 Neon, and made Signals and streaming detections generally available.

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

01 **Integration workflow enhancements for Notion, incident.io, SentinelOne**

Not checked yet IMPROVED 60

Expands integration workflows with filtered Notion database queries, team assignment for incident.io incidents, and SentinelOne passthrough alert routing.

– Names three distinct integration-specific enhancements v0.81.0

THINNER COVERAGE BELOW

02 **New AI model and fallback support** Not checked yet NEW 55

Adds `GPT-6.2 Astra`, `GPT-6.2 Sol`, and Gemini 4 Neon model support, plus custom model endpoints and organization-selected fallback models.

– Names specific models but no config steps shown v0.81.0

03 **New integrations and managed log ingestion sources**

Not checked yet NEW 53

Adds integrations for Cyberhaven, `iVerify`, UpGuard, Airtable, Raindrop, `MarkMonitor`, `OneLogin`, Thinkst Canary, and Cato Networks, plus managed log ingestion for Linear, Cursor, `incident.io`, Iru, and Generic JSON or OCSF data in Azure Blob Storage.

– Lists many named integrations but no setup mechanism given v0.81.0

04 **Expanded MCP detection authoring actions** Not checked yet IMPROVED 50

Expands Cotool MCP detection authoring with actions to create, test, and update detection code.

– Names specific authoring actions as a starting point v0.81.0

05 **Alert page severity sorting and code detection details**

Not checked yet IMPROVED 40

Adds severity sorting across pages and source integration and table details for code detections to alerts.

— Names alert page additions without further mechanism v0.81.0

06 **Signals and streaming detections generally available**

Not checked yet IMPROVED 30

Makes Signals and streaming detections generally available for real-time detection workflows.

— Announces GA status without describing mechanism v0.81.0

07 **Guided first-run setup for new organizations** Not checked yet

NEW 30

Adds a guided first-run setup to help new organizations connect integrations and complete their initial security configuration.

— Describes purpose but not the setup steps themselves v0.81.0

08 **Managed log ingest billing visibility** Not checked yet NEW

25

Adds managed log ingest usage, allowances, and alerts to billing.

— Bare description with no detail on thresholds or UI v0.81.0

WAS THIS USEFUL?  

Sourcegraph Amp

Feedback ⓘ announced · Nov 21, 2026 BLOG ↗

RANK

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

> Amp introduces a hands-free car mode for talking to its Puck agent while driving.

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

01 **Hands-free car mode for Puck agent** Not checked yet NEW

55

Amp adds a car mode that lets you talk to the Puck agent hands-free, automatically activated on the iPhone and Android apps when driving, or accessible at `ampcode.com/car` on any in-car web browser such as Tesla's.

— Names URL and trigger conditions but no mechanism detail. launch-20261121-4038b1ff

WAS THIS USEFUL? 👍 👎

◆ Image & Video Generation

Comfy Org ComfyUI

Feedback ⓘ announced · Nov 23, 2026

BLOG ↗ RANK

ComfyUI is a node-graph interface, API, and backend for running diffusion models, chaining generation steps into reusable visual workflows.

> Comfy Desktop 1.2 turns the desktop app into a front-end for quick image tasks, adding a built-in Apps launcher, a build management page with snapshot rollback, and automatic model/folder sharing across builds.

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

01 **Apps section with six one-click image tools** Not checked yet NEW

70

Comfy Desktop now hosts a native Apps section with six first-party one-click tools — Remove Background, SDR to HDR, Product Ads, Extend the Picture, Character Swap, and Product Swap — that run on a local GPU or Comfy Cloud without opening the node graph.

Remove an image background without touching the node graph — open the Apps section and run the built-in tool.

> 📍 In Comfy Desktop 1.2, go to Apps › Remove Background, drop in your image, and press Run.

— Names all six tools and run targets; gives a clear UI path launch-20261123-b1c54962

02 **Build details page with snapshots and rollback** Not checked yet

NEW

70

Each build now has a details page where you can update ComfyUI, set startup arguments, take snapshots, export snapshots, and manage folders, replacing manual file editing. A broken update can be rolled back by restoring a pre-update snapshot from this page.

Roll back a broken ComfyUI update by restoring a pre-update snapshot from the build details page.

> 📍 In Comfy Desktop 1.2, open the build details page for the affected install, select the snapshot taken before the update, and choose Restore.

— Describes mechanism and before/after, with a concrete rollback walkthrough

launch-20261123-b1c54962

03 **Shared model and folder sync across builds** Not checked yet

IMPROVED

60

Models, inputs, and outputs can now be shared automatically across all builds, removing the need to manually edit `extra_model_paths.yaml` or copy files between installs.

— Names the replaced config file but no exact new command launch-20261123-b1c54962

THINNER COVERAGE BELOW

04 **Enterprise shared builds from Comfy Cloud** Not checked yet NEW

30

Enterprise teams can now download and install builds shared from the Comfy Cloud platform so every team member runs the same setup.

— Only prose description, no steps or config named launch-20261123-b1c54962

WAS THIS USEFUL? 👍 👎

Higgsfield

Feedback ⓘ announced · Nov 21, 2026 BLOG ↗ RANK

Higgsfield is a generative video and image platform for creators, with a developer API.

> Higgsfield launched two new creative workspaces: Property Studio, which turns a floor plan into a narrated 3D walkthrough video, and Editing Room, a video-to-video editor with six targeted editing tools for footage up to ten minutes long.

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

01 Editing Room video-to-video editor with six edit tools

Not checked yet NEW 90

Editing Room is a video-to-video workspace that accepts uploads up to ten minutes long, automatically splits them into shots, and applies edits to selected moments without re-uploading. It adds Swap person (powered by Genjutsu V1; multiple shots, segments 2 to 30 seconds, optional reference images, prompt toggle), Change angle (front, profile, top-down, wide, default, close-up; within one shot, 2 to 15 seconds), Change location (multiple shots, segments 2 to 30 seconds, reference image, optional prompt), Move camera (dolly, pan, or orbit; within one shot, 2 to 15 seconds), Relight (within one shot, 2 to 15 seconds; presets such as Hard neutral split or Golden hard Rembrandt, custom light direction/color controls, or a text prompt), and Edit with prompt (paint over a selected area via the Brush panel with adjustable brush size, up to 30 seconds within one shot). Multiple edits can be stacked on the same moment and run in parallel, tracked by a Generating (N) indicator; results can be reviewed via Split, Compare (movable slider), or Final preview modes and a History panel. Output resolution (480p, 720p, or 1080p) and a Batch size setting control generation, with credit cost shown on the Generate button; finished videos export as MP4 and auto-save to the My videos tab.

Replace a person across several shots and then relight the same segment — without re-uploading the video.

📍 1. In Editing Room, select Upload video to edit and choose your video (up to 10 minutes). 2. Click the lane above the timeline to add an edit; drag the edges to cover 2–30 seconds across the shots you want. 3. In the left panel, choose Swap person; add a reference image with '+' and describe the replacement in the Prompt field. 4. Set Quality to 1080p and Batch size to 1, then click Generate. 5. With the same segment still selected, choose Relight; pick a Preset such as Golden hard Rembrandt and click Generate. 6. Click Compare to review both edits side-by-side with the vertical slider, then click Export to download the finished MP4.

Inpaint a specific object in a single shot using a painted mask and a text prompt.

📍 1. Open your project in Editing Room and click the timeline lane to select a 2–30-second segment within one shot. 2. In the left panel, choose `Edit with prompt` to open the Brush panel. 3. Paint over the area to change; adjust `Brush size` as needed. 4. In the text field below, describe what should appear (e.g., 'a red fire extinguisher'). 5. Set `Quality` to 720p, check the credit cost on the Generate button, then click Generate. 6. Open `History` to compare versions; use `Split` view to see the original stacked above the result.

— Names every tool, parameter range, and preset; UI-driven but no API endpoint given.

`launch-20261121-226af289`

---

02 **Property Studio floor-plan-to-video walkthroughs** `Not checked yet`

`NEW` 75

Property Studio converts a floor plan into a furnished 3D layout and a narrated video tour. Users set video duration (45 seconds to 4 minutes), aspect ratio (16:9, 9:16, 4:3, or 3:4), ceiling height in meters or feet, and resolution (720p, 1080p, or 4K) before generating; up to 20 reference images per project can be added and tagged as `Render`, `Reference`, or `Global style`. Narration can come from a library voice, an uploaded voice sample (15 seconds to 2 minutes), or a ready-made recording, and the auto-generated room-by-room script can be reviewed, regenerated, or replaced before final generation. A real-time `Credits used` counter tracks consumption, with walkthroughs averaging 1,500 to 2,000 credits depending on duration and complexity.

Create a narrated video tour from a floor plan for a property listing, specifying resolution and aspect ratio suited to mobile sharing.

📍 1. Go to higgsfield.ai/property-studio and click `Create walkthrough`. 2. Upload your floor plan (PNG, JPEG, WebP, or PDF, up to 15 MB). 3. Set Duration to `45 seconds`, Aspect ratio to `9:16`, Resolution to `1080p`, and enter the ceiling height. 4. Click `Create walkthrough` to start the project. 5. On `Rooms & references`, verify room names and add up to 20 reference images, tagging each as `Render`, `Reference`, or `Global style`. 6. Review the generated 3D layout and describe any changes in the text field, then click `Next step`. 7. Edit the auto-generated script, then click `Choose a voice` and select a library voice, upload a 15-second to 2-minute voice sample, or upload a ready recording. 8. Preview and click `Generate` to produce the final video tour.

— Full UI workflow and parameters given, but no API/CLI surface named.

`launch-20261121-0d71b943`

WAS THIS USEFUL?  

Runway

Feedback ⓘ Minimax H3 Image NOTES ↗ RANK

Runway is an AI video and image generation platform with models for text-to-video, image-to-video and editing, available in a web app and through an API.

> Runway expanded its model lineup with five new image/video generation models — Minimax H3 Image, Ideogram 4.6, Kling 4.0 Image, LTX Alphagen, and Seedance 2.5 Flash — and extended its MCP integration for ChatGPT and Claude with 3D model generation and a conversational Ad Director workflow.

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

01 **Ad Director in MCP for ChatGPT** Not checked yet NEW 76

Runway's MCP for ChatGPT now includes Ad Director, which takes a creative brief and product reference images, generates an ad plan for review, and produces the final product ad entirely within the ChatGPT conversation.

Create a product ad end-to-end by briefing Ad Director inside ChatGPT — useful when you want to iterate on the ad plan before committing to generation.

📍 Steps: 1. Open ChatGPT and connect the Runway MCP server (runway.com/mcp/chatgpt). 2. In the chat, provide Ad Director with a creative brief and attach your product reference images. 3. Review the generated ad plan and ask follow-up questions or request adjustments. 4. Approve the plan to let Ad Director produce the final product ad, previewing results inline.

— Exact connection steps and conversational workflow provided snapshot-20261124

02 **Minimax H3 Image model with high-speed mode** Not checked yet NEW 72

Runway added the Minimax H3 Image model, which accepts up to 16 reference images per generation. An optional high-speed mode can be enabled in settings for faster generation.

Use Minimax H3 Image with high-speed mode to generate images faster when working with multiple references.

📍 In the Runway workspace, select the Minimax H3 Image model, upload up to 16 reference images, then go to Settings and enable High-speed mode before generating.

— Names model, reference limit, and exact UI toggle with steps Minimax H3 Image

03 **Seedance 2.5 Flash video tier** Not checked yet NEW 70

Runway added Seedance 2.5 Flash, a faster and cheaper video generation tier with native audio at 480p or 720p resolution. A single generation can use up to 10 image references, 5 video references, and 5 audio references, and the tier now supports edit and extend operations on generated video.

— Thorough specs and references but no direct usage example Seedance 2.5 Flash

04 **Kling 4.0 image model** Not checked yet NEW 66

Runway added the Kling 4.0 model for generating images from text or image prompts, supporting up to 10 reference images and output at 2K, 4K, or 8K resolution across 8 aspect ratios.

— Rich numeric specs but no usage walkthrough Kling 4.0 Image

05 **Ideogram 4.6 image model** Not checked yet NEW 64

Runway added Ideogram 4.6, a new image model supporting text-to-image generation, image-to-image generation, precise mask edits, and strong text rendering at 2K or 4K resolution across aspect ratios from 1:6 to 6:1.

— Detailed capabilities and limits but no usage steps given Ideogram 4.6

06 **LTX Alphagen matte and alpha channel model** Not checked yet NEW 62

Runway added LTX Alphagen, a third-party model that converts any video into a production-grade matte with a separate alpha channel cutout — the first Runway model to return two simultaneous outputs. ProRes is now supported as an output format through this model.

— Explains unique dual-output mechanism and named output format LTX Alphagen

THINNER COVERAGE BELOW

07 **3D model generation via MCP in ChatGPT and Claude** Not checked yet NEW 47

Runway's MCP now supports generating 3D models from text prompts or images directly inside ChatGPT or Claude, with inline model exploration and iteration without leaving the chat.

— Describes capability and surfaces but no concrete command 3D Model Generation in MCP

NEW INTEGRATION Google Display campaign publishing

NEW INTEGRATION Google Performance Max campaign publishing with performance tracking

> Google Display and Performance Max in Runway Ads NOTES ↗ STABLE

> Runway Ads adds Google Display and Performance Max publishing, with support for carousel, multi-image, multi-video, and no-CTA ads, plus daily performance tracking and pause/resume.
>
> Publish Runway-made creatives directly to Google Display campaigns, in addition to the existing Demand Gen support. Not checked yet
>
> Publish Runway-made creatives directly to Google Performance Max (PMax) campaigns with daily performance tracking and pause/resume controls. Not checked yet
>
> Runway Ads now supports carousel ads, multi-image ads, and multi-video ads for Google campaign publishing. Not checked yet
>
> Runway Ads now supports ads without a call to action for Google campaign publishing. Not checked yet

---

WAS THIS USEFUL? 👍 👎

◆ Coding Agent Add-ons

Token Optimizer

Feedback ⓘ v5.13.76 NOTES ↗ RANK

Token Optimizer is a coding-agent plugin that trims wasted context, checkpoints work across sessions and compactions, and reports token spend on a live dashboard.

> Token Optimizer shipped two new slash commands for prioritising and safely applying context cleanups, plus automatic session caching and upfront cost previews when resuming large idle sessions.

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

01 **Prioritised action list via /token-optimizer:next** Not checked yet

NEW

70

`/token-optimizer:next` shows a prioritised action list, including savings already earned and the top remaining actions ranked by real monthly value.

After working for a while, run this to see which optimisations would save the most money this month before acting on any of them.

```
$ /token-optimizer:next
```

— Runnable command described, but no numeric savings example given. v5.13.76

02 **/token-optimizer:apply-safe cleanup command** Not checked yet NEW

70

`/token-optimizer:apply-safe` cleans up duplicate context entries and unused startup load in a single approved step, with full undo available if the change needs reverting.

Remove duplicate context entries and unused startup load in one step, with a full undo available if anything looks wrong.

```
$ /token-optimizer:apply-safe
```

— Runnable command with clear behaviour and undo path. v5.13.76

THINNER COVERAGE BELOW

03 **Session resume caching and cost preview** Not checked yet NEW

50

Large paused sessions now stay cached automatically whenever a refresh is cheaper than a full rebuild, subject to hard daily limits. When resuming a large idle session, the cost is now shown upfront with options to start fresh, compact, or continue.

— Explains mechanism and options but no config surface or command named. v5.13.76

WAS THIS USEFUL? 👍 👎

AGENT

◆ Web Search & Scraping for Agents

Perplexity API

Feedback ⓘ updated · Nov 24, 2026 NOTES ↗

RANK

Perplexity API provides language models with real-time web search and retrieval for building applications that research and answer questions.

> Perplexity added support for a new model, google/gemini-4-neon, to its Agent API with per-token pricing.

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

01 Gemini-4-Neon support in Agent API

Not checked yet NEW

55

The Agent API can now call `google/gemini-4-neon`, priced at $3.00 per million input tokens, $0.15 per million cached input tokens, and $15.00 per million output tokens (reasoning tokens included).

— Names model and pricing but no usage example or call syntax snapshot-20261124

WAS THIS USEFUL? 👍 👎

◆ AI Agent Frameworks

Nous Research Hermes

Feedback ⓘ v0.23.4 NOTES ↗

CODE ↗ RANK

Hermes is a terminal-native, self-improving AI agent for autonomous coding and tasks, with persistent memory, agent-created skills, and a multi-platform messaging gateway.

> Hermes v0.23.4 lets plugins take full control of a profile's backend in an isolated sandbox, with the host standing aside across all entry points, and adds a large batch of new plugin-catalog integrations.

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

01 **Plugin-controlled profile backends with host stand-aside**

Not checked yet NEW 57

A plugin can run a profile's backend in an isolated sandbox, and while it does, all host services and entry points skip that profile — including `cron`, messaging platforms, Kanban, Webapp, CLI, TUI, and desktop build agents.

— Explains mechanism and lists affected surfaces but no config/flag to invoke it v0.23.4

02 **New plugin-catalog entries** Not checked yet NEW 45

Adds a large batch of new plugin-catalog entries including `slack-history`, `utcp-bridge`, `katara` memory provider, `polytropos-router`, `rollstack`, `aisecuritywarden`, `oxylabs-web-api`, `beeper-platform`, `valetudo`, `abliteratedllm` model provider, `hermes-island`, and the `HybridOps` recovery plugin, among others.

— Names many plugins but gives no detail on what each does v0.23.4

WAS THIS USEFUL? 👍 👎

CrewAI

Feedback ⓘ 1.15.45 NOTES ↗ RANK

CrewAI is an open-source Python framework for building multi-agent workflows using role-based AI agent Crews and event-driven Flows.

> CrewAI added the ability to save agent task output as a PDF artifact generated from HTML or Markdown content.

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

01 **PDF export for task output** Not checked yet NEW 35

Agent task output can now be saved as a PDF artifact generated from HTML or Markdown content.

— Names format conversion but no API or flag given 1.15.45

WAS THIS USEFUL? 👍 👎

PydanticAI

Feedback ⓘ v2.73.0 NOTES ↗ RANK

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

> PydanticAI's v2.73.0 release adds a runtime deprecation warning for stale Anthropic/OpenAI models, new GPT-Live voice session controls, deferred tool-call delegation between SubAgents, and image-size controls in Coder, alongside breaking changes to FallbackModel behaviour and several dependency floors.

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

01 **Coder image dimension control** Not checked yet NEW 85

The new FileSystem(max_image_side=...) argument on Coder controls maximum image dimensions and downscales images to at most 1024 pixels on the longest side by default. The `coder` extra now requires `pillow>=12.3.0` to support this.

Cap the size of images the Coder reads from disk to avoid sending oversized payloads to the model.

```python
from pydantic_ai_harness import Coder
from pydantic_ai_harness.filesystem import FileSystem

agent = Agent(
    'anthropic:claude-fable-5-2',
    capabilities=
[Coder(filesystem=FileSystem(max_image_side=512))],
)
```

— Runnable example and exact config key with default value v2.73.0

02 **FallbackModel no longer treats unsupported settings as a trigger**

Not checked yet 75

BREAKING

A `FallbackModel` that previously relied on a 400 error (from sending unsupported unified settings to Claude, Gemini, or Grok) to fall through to a later model now stays on the first model without the setting instead. To preserve the old behaviour, list a model that supports the setting first.

— Clear before/after and a migration note, no runnable snippet v2.73.0

03 **Runtime warning for deprecated models** Not checked yet NEW

70

PydanticAI now raises a `DeprecatedModelWarning` at runtime when a request targets a deprecated Anthropic or OpenAI model, letting callers catch the warning (e.g. via `warnings.filterwarnings`) before the provider itself errors out.

Catch a warning before a deprecated model is called, so a CI step or test suite surfaces the problem immediately rather than waiting for a provider-side error.

python

```python
import warnings
from pydantic_ai.exceptions import DeprecatedModelWarning

warnings.filterwarnings('error', category=DeprecatedModelWarning)

agent = Agent('openai:gpt-4')
result = agent.run_sync('Summarise this report.')
```

— Named warning class with a runnable catch example v2.73.0

THINNER COVERAGE BELOW

04 **GPT-Live voice session controls** Not checked yet NEW 55

`RealtimeSession.append_voice_instructions` appends dynamic voice instructions mid-session on GPT-Live, with a Twilio Media Streams bridge example provided; a custom voice object can now also be passed via `openai_voice`.

— Named method and parameter but no runnable snippet included v2.73.0

05 **Agent.override capability control and public instruction types**

Not checked yet NEW 55

`capabilities` can now be passed to `Agent.override` to control agent capabilities at override time, and the previously-private `AgentInstructions` and `SourcedInstruction` types are now importable. The private `Agent._override_root_capability` context variable now holds a (capability, injected) tuple, breaking code that reads it directly.

— Names the API surfaces and the internal breaking change, no example v2.73.0

06 **Dependency floor bumps for openai and boxd** Not checked yet

55

BREAKING

The `openai` floor is raised from 3.31.0 to 3.50.0 on every extra that installs `openai`, and `pydantic-ai-harness[boxd]` now requires `boxd>=0.3.3,<0.4`, breaking code that calls 0.2-only boxd APIs directly.

— Exact version numbers given with a migration implication v2.73.0

07 **Thinking profiles for six model families** Not checked yet NEW

50

Added thinking profiles for the Nemotron, Inkling, MiMo, Step, Hy, and Ling model families on OpenAI-compatible gateways.

— Names six models but no mechanism or usage shown v2.73.0

08 **Deferred tool-call delegation in SubAgents** Not checked yet NEW

45

`SubAgents` can now pass a delegate's deferred tool calls back to the caller and resume the delegate once those calls are answered.

— Mechanism described but no example or config shown v2.73.0

09 **OpenAI Responses input_transformations surfaced** Not checked yet

NEW

45

`input_transformations` from the OpenAI Responses API is now surfaced in `provider_details` and on the model request span.

— Names fields but gives no usage context v2.73.0

└──▷ **BREAKING ON UPGRADE**

! The `openai` floor is raised from 3.31.0 to 3.50.0 on every extra that installs `openai`.

! The private `Agent._override_root_capability` context variable now holds a (capability, injected) tuple; code that reads it directly needs updating.

! The `coder` extra now requires `pillow>=12.3.0`.

! A `FallbackModel` that relied on a 400 error (from sending unsupported unified settings to Claude, Gemini, or Grok) to reach a later model now stays on the first model without the setting; list a model that supports the setting first to preserve the behaviour.

! `pydantic-ai-harness[boxd]` now requires `boxd>=0.3.3,<0.4`; code that calls 0.2-only boxd APIs directly needs updating.

WAS THIS USEFUL?  

OpenAI Agents SDK

Feedback ⓘ v0.26.0 NOTES ↗ RANK

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

> OpenAI Agents SDK v0.26.0 tightens the experimental Codex tool's image handling and adds strict JSON parameter validation, gives MCP call metadata top priority, and ships several breaking changes around Docker session mounts and custom trace providers.

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

01 Codex tool named image-attachment control

Not checked yet

95

BREAKING

The experimental Codex tool now requires passing named image references via `image_attachments`, with those names used as the `local_image` path in calls; a model-supplied file path no longer grants access to local images. This is a breaking change — existing calls that rely on direct path access will be rejected, so callers must grant a named reference via `image_attachments` for each trusted image.

— Names exact flag, path field, and gives full migration mechanism. v0.26.0

02 Docker rejects InContainerMountStrategy sessions

Not checked yet

90

BREAKING

Docker now rejects `InContainerMountStrategy` (including read-only mounts) during session creation and resume; replace with `DockerVolumeMountStrategy` and start a fresh session — existing sessions using `InContainerMountStrategy` cannot be reused.

— Names exact strategies and gives concrete migration steps. v0.26.0

03 Custom TraceProvider must implement is_disabled()

Not checked yet

75

BREAKING

Custom `TraceProvider` implementations must implement is_disabled() to return their actual disabled state; in v0.25.0 Realtime ignored this method, so custom providers that do not implement it will now have server traces silently suppressed.

— Names exact method and prior version regression, no code sample. v0.26.0

04 Strict JSON validation mode for Codex tool parameters

Not checked yet NEW

65

The Codex tool can now enable strict JSON validation for its parameters by setting `parameter_validation_mode="json"`; the default remains `"python"`.

— Exact config value given but no deeper mechanism explained. v0.26.0

05 **MCP call metadata priority over resolver defaults** Not checked yet

IMPROVED 60

Explicit call metadata now takes priority over `tool_meta_resolver` and then discovery defaults for conflicting MCP keys.

— Names the resolver and precedence order but no usage example. v0.26.0

THINNER COVERAGE BELOW

06 **Realtime turn-end guardrail checks on transcripts** Not checked yet

IMPROVED 45

Realtime now checks remaining text and audio transcripts against output guardrails at turn completion.

— Describes behavior but no config surface or API named. v0.26.0

07 **Redacted call arguments retained in handoff session history**

Not checked yet IMPROVED 30

Handoff filters now keep redacted call arguments in saved session history.

— Brief behavioral note without mechanism or config detail. v0.26.0

08 **GPT-Live demo standby and citations** Not checked yet NEW

30

The GPT-Live browser demo now adds speech-triggered standby and search citations.

— Demo-only description with no API or config surface. v0.26.0

└──▷ **BREAKING ON UPGRADE**

! Docker now rejects `InContainerMountStrategy` (including read-only mounts) during session creation and resume; replace with `DockerVolumeMountStrategy` and start a fresh session — existing sessions using `InContainerMountStrategy` cannot be reused.

! Custom `TraceProvider` implementations must implement is_disabled() to return their actual disabled state; in v0.25.0 Realtime ignored this method, so custom providers that do not implement it will now have server traces silently suppressed.

! In the experimental Codex tool, a model-supplied file path no longer grants access to local images; grant a named reference via `image_attachments` for each trusted image instead — existing calls that rely on direct path access will be rejected.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.9.87 NOTES ↗ RANK

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

> HyperFrames expanded Studio's sealed preview workflow, letting embedding apps style, read, and edit sealed previews without breaking their seal.

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

01 **Sealed preview styling, editing and markup access in Studio**

Not checked yet NEW 42

Apps embedding Studio can now style a sealed pick to match Studio's own panel appearance, read a sealed preview's page markup for programmatic access to its HTML structure, and edit text in place on a sealed preview's canvas without unsealing the preview first.

— Describes what changed but no API names or mechanism given v0.9.87 v0.9.85 v0.9.86

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ ai@7.0.181 NOTES ↗ RANK

The Vercel AI SDK provides TypeScript APIs for model generation, structured output, tool use, and streaming application interfaces.

> Vercel AI SDK 7.0.181 adds visibility into agent tool-loop termination and prompt conversion warnings, and tightens telemetry extraction to drop provider options from captured text and file parts.

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

01 `stoppedBy` field on generation results

Not checked yet NEW

72

`generateText` and `streamText` results, the `onEnd` callback, and the finish part now expose a `stoppedBy` field so callers can inspect why a tool loop stopped — for example distinguishing a `maxSteps` limit from a natural finish.

Check why an agent tool loop terminated — e.g. to distinguish a max-steps limit from a natural finish — after calling generateText.

typescript

```typescript
import { generateText } from 'ai';

const result = await generateText({
  model: 'openai/gpt-6-astra',
  prompt: 'Solve this step by step.',
  maxSteps: 5,
  onWarning: (warning) => console.warn('Warning:', warning),
});

console.log('Stopped because:', result.stoppedBy);
```

— Names exact fields and surfaces; runnable example provided ai@7.0.181

02 Per-call `onWarning` callback

Not checked yet NEW

62

A new per-call `onWarning` callback can be attached to `generateText` or `streamText` to receive prompt conversion warnings alongside step warnings.

Check why an agent tool loop terminated — e.g. to distinguish a max-steps limit from a natural finish — after calling generateText.

```typescript
import { generateText } from 'ai';

const result = await generateText({
  model: 'openai/gpt-6-astra',
  prompt: 'Solve this step by step.',
  maxSteps: 5,
  onWarning: (warning) => console.warn('Warning:', warning),
});

console.log('Stopped because:', result.stoppedBy);
```

— Clear callback name and attach points, example shows usage `ai@7.0.181`

THINNER COVERAGE BELOW

03 **Telemetry extraction strips provider options** Not checked yet

IMPROVED 43

Telemetry extraction now strips provider options from text and file parts while still preserving arbitrary JSON state and the options actually sent to the provider.

— Behavior change described but no API or example to act on `ai@7.0.181`

WAS THIS USEFUL? 👍 👎

CopilotKit

Feedback ⓘ v1.90.1 NOTES ↗ RANK

CopilotKit connects AI agents to application interfaces with shared state, tools, and generative UI components.

> CopilotKit v1.90.1 adds fullscreen display for MCP Apps widgets and changes runtime thread history and debug event handling in a way that breaks existing setups.

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

01 **Fullscreen support for MCP Apps widgets** Not checked yet NEW

30

MCP Apps widgets now support fullscreen display.

– States the capability but no mechanism or usage detail v1.90.1

02 **Runtime thread history and debug event handling changed**

Not checked yet

20

BREAKING

Runtime thread history and debug event handling have changed; existing setups relying on the previous behavior will break on upgrade.

– No migration steps or specifics on what changed v1.90.1

└──▷ **BREAKING ON UPGRADE**

! Runtime thread history and debug event handling have changed — existing setups relying on the previous behavior will break on upgrade.

WAS THIS USEFUL? 👍 👎

trigger.dev

Feedback ⓘ v4.8.10 NOTES ↗ RANK

Trigger.dev – build and deploy durable AI agents and workflows

> trigger.dev expanded its remote MCP server with webhook and TRQL-based alert tools, added configurable ClickHouse task-event retention (a breaking change to the default retention window), and rolled out new deployment and schedule dashboards alongside audit log filtering, summaries, and OpenTelemetry metrics export.

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

01 **MCP server webhook and alert management tools** Not checked yet

NEW 90

The remote MCP server's `create_alert` tool can create alerts on custom TRQL queries via the `custom.query` capability with a `query` and a `signal`. The server can also search webhook endpoints by subscriber, source, status, and text; show a webhook task with its subscribed endpoints; list runs waiting on an endpoint; remove a signing secret; generate a verification token; pause or resume an endpoint; and cancel a run's wait on it.

– Names every tool and parameter but not an invocation example v4.8.9

02 **Configurable ClickHouse task event retention** Not checked yet

90

BREAKING

Self-hosted ClickHouse deployments can now control how long task events are retained via the `EVENTS_CLICKHOUSE_RETENTION_DAYS` environment variable. This changes default behavior: new task events are now deleted after about 37 days instead of about a year; events stored before upgrading keep their original expiry.

– Exact env var, old/new retention numbers, and upgrade note v4.8.9

03 **OpenTelemetry metrics export from instrumentations**

Not checked yet IMPROVED 83

Metrics from instrumentations declared in `trigger.config.ts` (such as HTTP, undici, and pg) are now recorded and exported instead of silently dropped. After upgrading, remove any manual instrumentation.setMeterProvider(metrics.getMeterProvider()) call from `init`.

Remove the now-redundant manual meter-provider binding in your instrumentation init after upgrading so you do not register duplicate observable callbacks.

```typescript
// In trigger.config.ts — remove this line from your instrumentation's init():
// instrumentation.setMeterProvider(metrics.getMeterProvider());
```

— Exact config file, method call, and migration step provided v4.8.10

04 **Deployment activity grid and detail pages** Not checked yet NEW

75

The dashboard's deployments list now includes an Activity switch that displays a year of deploys as a day-by-day grid showing daily deploy counts, failed-deploy markers, weeks shipped, and longest gap. Each deployment now has a full detail page with an overview of tasks and runs, build logs, an activity timeline, and filtering by status, branch, and version, with rollback showing which tasks the older version is missing.

— Describes new UI surfaces but only a navigation path, not a config v4.8.9

05 **Audit log filtering and per-key activity summaries**

Not checked yet NEW

75

The audit events API can now be filtered by request source and target name via `filter[source]` and `filter[targetName]`. A new audit log page in organization settings lets you filter credential and member-change history by time, category, event, actor, key, `filter[source]`, project, environment, and target, and shows a summary of activity for a single key or token — first and last seen, event count, number of IPs and countries, and for organization keys, last-used and revocation timestamps.

— Named filter params and a reachable settings page v4.8.10

06 **CLI deprecation warnings for trigger deploy** Not checked yet NEW

75

`trigger deploy` now warns when your CLI version will soon stop being supported for new deploys, showing the last day it still works and the upgrade command. The warning appears on every build path, including `--native-build`, `--local-build`, and `--depot-build`, and never blocks a deploy.

— Named flags and behavior, but no exact warning text v4.8.10

07 **(improved) Runs list API and bulk action improvements**

Not checked yet IMPROVED

73

The runs list API now returns an optional `appliedFilters` field in `ListRunResponse` showing the `createdAt` window used, so callers can tell when the default 7-day period applied; bulk replay now lets you choose the destination version and environment, bulk actions on selected runs now include runs older than 7

days, and an unreadable URL filter is now shown as a warning instead of clearing all filters.

– Named field and response type with its purpose Three concrete UI behavior changes with no named config

v4.8.9

08 **Per-schedule dashboard pages with run previews** Not checked yet

NEW

65

Each schedule now has its own dashboard page with its runs and an Upcoming tab showing a calendar of future runs or an hourly/daily rhythm view for high-frequency schedules. Creating and editing a schedule now previews its next runs beside the form.

– Clear UI description but only a navigation starting point v4.8.9

THINNER COVERAGE BELOW

09 **Plain-text CLI output in CI** Not checked yet IMPROVED 50

CLI spinners now print plain text in CI and when stdout is piped or redirected, so `trigger deploy` logs in GitHub Actions and other CI systems no longer contain spinner frames or cursor escape codes.

– Clear fix described but minor scope and no config surface v4.8.9

└──▷ **BREAKING ON UPGRADE**

! Self-hosted ClickHouse deployments: new task events are now deleted after about 37 days instead of about a year. Set `EVENTS_CLICKHOUSE_RETENTION_DAYS` to extend retention; events stored before upgrading keep their original expiry.

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ harness-python/v0.5.0

NOTES ↗ RANK

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

> Strands Agents added runtime tool loading, deterministic tool ordering, OpenAPI metadata publishing, MCP session auto-reconnect, and agent-level background task cancellation, while also shipping two breaking changes around default model selection and context-manager offload behavior.

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

01 Auto-reconnect for MCP client sessions

Not checked yet NEW

75

The Python MCPClient now supports an `auto_reconnect=True` option so a long-running agent can keep a dropped MCP server session alive mid-conversation, used via `async with client:` and await client.list_tools().

Keep a long-running Python agent connected to an MCP server that may drop its session mid-conversation.

python

```python
from strands.tools.mcp import MCPClient

client = MCPClient(
    server_url="https://my-mcp-server/sse",
    auto_reconnect=True,
)
async with client:
    tools = await client.list_tools()
```

— Named flag with a runnable code example. python/v1.66.0

02 Cancel background tasks via agent.cancel

Not checked yet NEW

75

await agent.cancel(cancel_background_tasks=True) cancels the current agent turn along with any background tasks it spawned, instead of letting them run to completion.

Cancel an agent and all its in-flight background tasks at once instead of letting them run to completion.

```python
# cancel the agent turn AND any background tasks it spawned
await agent.cancel(cancel_background_tasks=True)
```

— Exact method and parameter shown with usage. python/v1.66.0

THINNER COVERAGE BELOW

03 Anthropic Bedrock client default model change

Not checked yet

NEW INTEGRATION BREAKING

45

The Anthropic Bedrock client now defaults to Claude Opus 5.5; agents that relied on the previous default model will switch models automatically on upgrade.

— Names the new default model, no migration steps given. python/v1.66.0

04 Dynamic tool loading in context manager

Not checked yet NEW

30

The context manager can now register tools at runtime, allowing tools to be loaded dynamically rather than only configured upfront.

— No flag, API, or example given for runtime registration. python/v1.66.0

05 Finer-grained context-manager offload conditions

Not checked yet

30

BREAKING

Context-manager offload conditions now decide at a finer granularity (Python); existing offload condition logic may behave differently after upgrading.

— Vague on what the finer granularity actually changes. python/v1.66.0

06 Deterministic tool spec ordering

Not checked yet IMPROVED

25

Tool specs can now be sorted to produce deterministic tool ordering.

— Single line, no flag or mechanism named. python/v1.66.0

07 OpenAPI metadata publishing for agents

Not checked yet NEW

25

Agents can now publish OpenAPI metadata from the site.

— No endpoint, format, or path specified. python/v1.66.0

NEW INTEGRATION SageMaker AI decision model integration

NEW INTEGRATION Dogwood authorization handler for interventions

NEW INTEGRATION ClickHouse Loom memory store catalog entry

└──▷ BREAKING ON UPGRADE

! The Anthropic Bedrock client now defaults to Claude Opus 5.5; agents that relied on the previous default model will switch models on upgrade.

! Context-manager offload conditions now decide at a finer granularity (Python); existing offload condition logic may behave differently after upgrade.

harness-python/v0.5.0 NOTES ↗ LATEST STABLE

Strands Agents adds dynamic tool loading in the context manager, `auto_reconnect` for MCP sessions, opt-in background-task cancellation, and a `SageMakerDecisionModel` for SageMaker AI endpoints.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch harness-python/v0.5.0
https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout harness-python/v0.5.0
```

Add `auto_reconnect` to MCPClient to automatically reconnect dropped MCP sessions (Python). Not checked yet

Add opt-in `cancel_background_tasks` parameter to agent.cancel() to also halt in-flight background tasks. Not checked yet

Add dynamic tool loading to the context manager. Not checked yet

Add a `reminder_interval` setting to the todos plugin. Not checked yet

Allow configuring the background-tasks management tool name. Not checked yet

+8 more

└──▷ BREAKING ON UPGRADE

! The context manager's `stash` operation is renamed to `archive` (TypeScript); code calling `stash` must be updated.

! The Bedrock client now defaults to Claude Opus 5.5 instead of the previous default model; agents that relied on the old default will switch models on upgrade.

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ langchain-classic==1.1.0 NOTES ↗ RANK

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

> LangChain's latest releases add a standard `reasoning_effort` parameter and configurable LangSmith tracer addresses to langchain-classic, thinking-level support and v2 content-block streaming to langchain-ollama, and new streaming/tracing checks to langchain-tests — while dropping Python 3.10 support across the classic and Ollama packages.

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

01 Standard `reasoning_effort` chat model parameter

Not checked yet

NEW

75

Any supported chat model can now take `reasoning_effort` as a standard parameter to cap reasoning cost, without needing a provider-specific wrapper — e.g. init_chat_model("openai:o3", reasoning_effort="low").

Cap reasoning cost on a chain call by setting `reasoning_effort` directly on the model — no provider-specific wrapper needed.

python

```python
from langchain.chat_models import init_chat_model

model = init_chat_model("openai:o3", reasoning_effort="low")
result = model.invoke("Explain RSA key exchange in one
paragraph.")
print(result.content)
```

— Runnable example given; mechanism and scope explained langchain-classic==1.1.0

02 Configurable LangSmith tracer address

Not checked yet NEW

75

`LangChainTracer` now accepts a custom `address`, letting traces be routed to a self-hosted or private LangSmith instance instead of the default endpoint.

Route LangSmith traces to a self-hosted or private LangSmith instance by supplying a custom address to `LangChainTracer`.

```python
from langchain.callbacks import LangChainTracer

tracer =
LangChainTracer(address="https://langsmith.internal.example.com")
model = init_chat_model("openai:gpt-4o")
result = model.invoke("Summarise this alert.", config=
{"callbacks": [tracer]})
```

— Runnable example shows exact parameter and usage `langchain-classic==1.1.0`

THINNER COVERAGE BELOW

03 **Thinking levels in Ollama models** Not checked yet NEW

55

Thinking levels are now supported in Ollama models via an upgraded `ollama>=0.6.3` floor dependency in `langchain-ollama`.

— Names dependency version but no code example `langchain-ollama==1.2.0`

04 **Python 3.10 support dropped** Not checked yet BREAKING

50

Python 3.10 support is removed from both `langchain-classic` and the `langchain-ollama` partner package; the minimum supported Python version is now 3.11.

— Clear before/after but no migration steps given `langchain-classic==1.1.0` `langchain-ollama==1.2.0`

05 **Streaming and tracing checks in standard test suite**

Not checked yet NEW

50

`langchain-tests` standard tests now read and validate `tool-call-delta` chunks during streaming to catch malformed tool-call streams, and forward LangSmith CI environment variables to traces so test runs appear in LangSmith automatically.

— Names both checks but no command or config shown `langchain-tests==1.2.0`

06 **Content-block-centric streaming v2 for Ollama** Not checked yet

IMPROVED

45

`langchain-ollama` now uses the content-block-centric streaming (v2) added to `langchain-core` for finer-grained stream handling in Ollama chains.

— Mechanism named but no usage example provided `langchain-ollama==1.2.0`

07 **Explicit prompt caching for OpenAI integration** Not checked yet

NEW

25

The OpenAI integration now supports explicit prompt caching.

— One-line mention with no mechanism or config shown `langchain-classic==1.1.0`

---

08 Package version tracking in Ollama tracing metadata

Not checked yet NEW 25

Tracing metadata for Ollama calls in LangSmith now includes package version tracking.

— Bare mention, no detail on how it's surfaced `langchain-ollama==1.2.0`

└──▷ BREAKING ON UPGRADE

! Python 3.10 support is removed from langchain-classic; the minimum supported Python version is now 3.11.

! Python 3.10 support is removed from the langchain-ollama partner package.

WAS THIS USEFUL? 👍 👎

◆ Agent Memory

Volcengine OpenViking

Feedback ⓘ v0.5.6 NOTES ↗ RANK

OpenViking is a context database that manages agent memory, knowledge retrieval, and skills.

> OpenViking v0.5.6 adds PostgreSQL-backed persistence for accounts, tasks, and watches to enable multi-node deployments without Redis, introduces a new `ov privacy run` command for secret-redacted subprocess execution, and overhauls Compile's persistent configuration model alongside a wide set of breaking changes to ACL, QueueFS, Gateway sessions, and content-write snapshot behaviour.

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

01 PostgreSQL persistence for accounts, tasks, watches

Not checked yet

95

BREAKING

Store accounts, tasks, and watch state in PostgreSQL by setting `account_store.provider`, `storage.task_store.provider`, and `storage.watch_store.provider` to `postgres`, enabling multi-process and multi-node deployments without Redis. Enabling postgres requires manually creating tables from the bundled SQL before starting the service, since the service will not run DDL or migrate existing File/MySQL/SQLite data; existing PostgreSQL Watch tables without account/user pagination indexes must have them added outside a transaction using `CREATE INDEX CONCURRENTLY` as described in `001_watch_tasks_postgres.sql`, and the service now validates schema on startup and refuses to start if the structure is incompatible.

Switch the task store to PostgreSQL for a multi-node deployment that must share task state without Redis.

yaml

```yaml
storage:
  task_store:
    provider: postgres
    dsn:
postgres://openviking:secret@db.example.com:5432/openviking
```

— Exact config keys, SQL file, and runnable example given v0.5.6

02 ov privacy run command for secret injection

Not checked yet NEW

90

Run external commands with secret injection via the new `ov privacy run` subcommand, which resolves `{{ov_privacy:...}}` references from env vars, env files, or CLI args and redacts sensitive values from the subprocess stdout and stderr. Set `OPENVIKING_PRIVACY_RUN_STRICT=1` to block argument expansion into subprocess argv; only owner-bound `{{ov_privacy:...}}` references are accepted.

Run a deployment script with injected secrets redacted from all subprocess output, blocking secret expansion into argv.

```
$ OPENVIKING_PRIVACY_RUN_STRICT=1 ov privacy run --env-file .env
-- ./deploy.sh
```

— Named flag, env var, and runnable command example v0.5.6

---

03 **Compile persistent configuration overhaul** Not checked yet

85

BREAKING

Manage Compile persistent configuration by caller-supplied, creator-scoped `compile_name`, with each run using an isolated task ID, frozen inputs, incremental baseline, and Git snapshot; complex-mode output now goes through an isolated Markdown file worker. This replaces the old operation interface: config queries no longer use `to`, snapshot versions are now Git commit IDs paginated with `before_snapshot`, and output now requires non-empty UTF-8 Markdown from Map/Shuffle/Reduce/Finalize workers writing to isolated directories; structured record/emit results, the old compile entry module, `resume`, and some artifact salvage paths have been removed. The old interface that accepted full inline input is temporarily retained but returns a deprecation notice.

— Extensive named fields and before/after but no runnable example v0.5.6

---

04 **Access level field in ls, tree, and stat responses**

Not checked yet

70

BREAKING

Read effective access level (`read` / `write` / `manage` / `denied`) directly from `ls`, `tree`, and `stat` responses via the new `access` field, so agents can gate operations before attempting them. Clients that perform strict field validation on these response models must accept the new field.

— Named endpoints and field with exact enum values v0.5.6

---

05 **DSH workspace memory can be fully disabled** Not checked yet

65

BREAKING

Disable DSH workspace memory entirely by setting `enabled: false` in the workspace config, which now skips recall, capture, commit, profile, and session initialization; projects that previously set this field but still generated memory will see changed behaviour after upgrading.

— Names config key and lists affected subsystems precisely v0.5.6

06 **Content-write snapshot opt-in and batch write options**

Not checked yet BREAKING 65

Content writes no longer create a snapshot by default; callers must pass `use_snapshot=true` to get snapshot behaviour. Batch writes now also accept `resolve_conflicts`, and single writes and the SDK accept `defer_processing`.

— Named parameters given but no usage example v0.5.6

07 **Gateway session soul.md injection and synchronous saving**

Not checked yet BREAKING 65

Inject a user's `soul.md` automatically when a Gateway session starts, with failures now surfaced explicitly rather than silently dropped. Gateway session saving now happens synchronously after each reply turn; the independent capture worker has been removed, so deployments that relied on its timing, internal queues, or old log patterns must update monitoring and troubleshooting rules.

— Behaviour change explained but no config surface named v0.5.6

08 **QueueFS Redis internal format change** Not checked yet BREAKING 65

QueueFS Redis internal format changed: `serial_key` and `group_key` are now persisted with messages; the Rust native extension and Python server must be upgraded together — do not run old and new QueueFS implementations side-by-side.

— Names exact persisted keys and upgrade constraint v0.5.6

THINNER COVERAGE BELOW

09 **Antigravity and Qoder coding-agent adapters** Not checked yet NEW 55

Add Antigravity App/CLI and Qoder CLI adapters to the installer with `--harness antigravity`, extending the set of supported coding-agent harnesses.

— Names the flag but little mechanism beyond that v0.5.6

10 **API error-handling changes for invalid input** Not checked yet 55

BREAKING

Invalid resource files now return HTTP 400; empty message parts are rejected before session creation; VLM content-filter responses are classified as safety errors. Clients that retry solely based on old HTTP status codes must review their retry logic.

— Names concrete status code and validation behaviour changes v0.5.6

---

11 **OpenCode v2 and OpenClaw integration improvements** Not checked yet

IMPROVED

50

OpenCode v2 now loads built-in Skills and injects context concurrently; OpenClaw recalls at the `before_prompt_build` stage to preserve prompt cache.

— Names a hook stage but no further mechanism v0.5.6

---

12 **ACL rejects cross-account grants** Not checked yet BREAKING

45

ACL now rejects grants to principals outside the resource's own account; existing cross-account ACL entries can no longer be maintained through the write interface.

— States the rule but no specific API or flag v0.5.6

---

13 **MCP retrieval tag filtering and Grep limit signal** Not checked yet

IMPROVED

40

Filter MCP retrieval results by tag and expose advanced keyword-search parameters; Grep now explicitly signals the caller when it hits its result limit.

— No parameter names or mechanism detail given v0.5.6

└──▷ BREAKING ON UPGRADE

! Compile API and data model: persistent config now uses a caller-supplied, creator-scoped `compile_name`; the run interface replaces the old operation interface; config queries no longer use `to`; snapshot versions are now Git commit IDs paginated with `before_snapshot`. The old interface that accepted full inline input is temporarily retained but returns a deprecation notice.

! Compile output now requires non-empty UTF-8 Markdown from Map/Shuffle/Reduce/Finalize workers writing to isolated directories; structured record/emit results, the old compile entry module, `resume`, and some artifact salvage paths have been removed.

! Content writes no longer create a snapshot by default; callers must pass `use_snapshot=true` to get snapshot behaviour.

! Enabling `account_store.provider: postgres`, `storage.task_store.provider: postgres`, or `storage.watch_store.provider: postgres` requires manually creating tables from the bundled SQL before starting the service; the service will not run DDL and will not migrate existing File/MySQL/SQLite data.

- ! Existing PostgreSQL Watch tables without account/user pagination indexes must have them added outside a transaction using `CREATE INDEX CONCURRENTLY` as described in `001_watch_tasks_postgres.sql`; the service validates schema on startup and refuses to start if the structure is incompatible.
- ! QueueFS Redis internal format changed: `serial_key` and `group_key` are now persisted with messages; the Rust native extension and Python server must be upgraded together — do not run old and new QueueFS implementations side-by-side.
- ! ACL now rejects grants to principals outside the resource's own account; existing cross-account ACL entries can no longer be maintained through the write interface.
- ! `ls`, `tree`, and `stat` responses now include an `access` field; clients that perform strict field validation on these response models must accept the new field.
- ! Gateway session saving now happens synchronously after each reply turn; the independent capture worker has been removed. Deployments that relied on the capture worker's timing, internal queues, or old log patterns must update monitoring and troubleshooting rules.
- ! `ov privacy run` only accepts owner-bound `{{ov_privacy:...}}` references; set `OPENVIKING_PRIVACY_RUN_STRICT=1` to prevent expansion into subprocess argv.
- ! DSH workspace `enabled: false` now fully takes effect, skipping recall, capture, commit, profile, and session initialization; projects that previously set this field but still generated memory will see changed behaviour after upgrading.
- ! Invalid resource files now return HTTP 400; empty message parts are rejected before session creation; VLM content-filter responses are classified as safety errors. Clients that retry solely based on old HTTP status codes must review their retry logic.

WAS THIS USEFUL?  

claude-mem

Feedback ⓘ v16.1.1 NOTES ↗ RANK

claude-mem gives coding agents persistent memory by capturing session activity, compressing it with a model, and injecting relevant context into later sessions.

> claude-mem introduces Cloud Sync, a paid add-on for persistent memory across devices, with a 30-day free trial and optional hosted memory generation.

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

01 **Cloud Sync for cross-device memory** Not checked yet NEW

65

A new Cloud Sync feature lets users connect an existing Cloud Sync plan to a new device during setup so memory follows them across machines. Setup also offers a 30-day free trial (card required, $0 today then $12/month) or an annual plan at $99 upfront, and once Cloud Sync is active users can enable optional hosted memory generation, with token usage from the original scan shown during setup.

— Pricing and setup flow given, but no config keys or commands v16.1.1

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

gram

Feedback ⓘ server@2.42.0 NOTES ↗ RANK

Gram provides MCP servers and AI-agent tools from APIs with access controls and observability.

> Gram's biggest addition this window is a full MCP connection health monitoring stack — a new API plus dashboard UI for connection logs, sessions, and sign-in analytics — alongside resource metadata/CSP support, deeper authorization-check visibility, and the graduation of two features out of feature-flag gating.

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

01 MCP connection health monitoring (API and dashboard)

Not checked yet NEW 88

The new `mcpConnectionHealth` API, gated behind the `mcp-connection-health` feature flag, returns paged connection logs (repeated attempts folded into one row), totals and per-filter-value counts, individual attempt step detail, lookup by reference, sign-in and renewal summaries, active sessions per day with average session length, and apps refused at sign-in; the flag can be targeted at individual people rather than the whole organization, and sign-in summaries can be filtered by step group via the optional `steps` parameter on `getSignInSummary` (matching the Connection Logs step filter). The dashboard surfaces this as an Overview tab with sign-in summary, denied clients, and recent sign-in activity, plus Connection Logs alongside Active Sessions with filters, reference search, per-attempt details, and a per-server view on each server's Clients and Sessions tab.

Retrieve a sign-in summary filtered to specific pipeline steps, matching what Connection Logs shows for those steps.

```javascript
getSignInSummary({ steps: ['sign-in', 'sign-in-renewal'] })
```

— Names endpoint, flag, params and UI tabs with a runnable example. server@2.42.0 dashboard@0.155.0

02 Resource metadata with Content Security Policy support

Not checked yet NEW 80

Resources can now carry a `meta` field passed to resource() or experimental_uiResource(), which is returned on `resources/list` and `resources/read`; MCP Apps UI resources can use this to declare a Content Security Policy via `ui.csp`, which the host enforces when rendering the resource.

Declare a Content Security Policy for an MCP Apps UI resource so the host enforces it when rendering the resource.

```javascript
experimental_uiResource({
  uri: 'myapp://dashboard',
  name: 'Dashboard',
  meta: {
    ui: {
      csp: "default-src 'self'; script-src 'self'"
    }
  },
  handler: async () => ({ html: '<h1>Dashboard</h1>' })
})
```

— Names exact functions, fields and endpoints with a working code sample.

`@speakeasy-api/functions@0.21.0`

03 **Authorization check visibility in Permissions and Platform MCP**

Not checked yet NEW 65

Upstream auth enforcement refusals — including personal connection misuse — can now be viewed under failed authorization checks on an agent's or person's Permissions tab and in Roles & Permissions, and listed per server via the Platform MCP `list_agent_authorization_checks` tool.

— Names UI location and tool but no example invocation shown. `server@2.41.0`

THINNER COVERAGE BELOW

04 **`merged_client_ids` field in auth provider consolidation preview**

Not checked yet NEW 45

Preview MCP Authentication Provider consolidations now include a `merged_client_ids` field listing which OAuth client IDs will merge.

— Names the field but gives no usage path or example. `server@2.41.0`

05 **Tool-name/kind block rule gap closed** Not checked yet IMPROVED 45

A tool-name or tool-kind block rule now also blocks calls where the tool name or kind cannot be determined, closing a gap where such calls previously fell outside the rule.

— Explains the fix's effect but not how to configure the rule. `server@2.41.0`

06 **GitHub Actions workload platform and prompt-based risk policies reach general availability**

Not checked yet IMPROVED 45

GitHub Actions is now available as a workload platform to every organization without requiring the `workload-platform-github-actions` feature flag, and prompt-based risk policies are now available to every project without a feature flag.

— Names the removed flag but no setup steps given. server@2.41.0

---

07 **Tool narrowing dialog deduplicates tools by annotation**

Not checked yet IMPROVED 25

The tool narrowing dialog on an MCP server's Team Access tab now counts each tool once, under the one annotation access is checked against.

— Names the UI location but describes only a minor counting fix. dashboard@0.154.2

---

08 **MCP Sessions graduates out of beta** Not checked yet IMPROVED 10

MCP Sessions is now out of beta.

— Bare status change, no mechanism or detail given. dashboard@0.155.0

WAS THIS USEFUL?  

◆ Agent Sandboxes

cplt

Feedback ⓘ 2026.11.24-020236-4ea5789 NOTES ↗ RANK

cplt is a kernel-level sandbox for AI coding agents, running Copilot CLI, Claude Code, OpenCode, Gemini CLI, and others with git and gh guards.

> cplt's biggest change this window is a managed config layer for organisation-wide sandbox policy, alongside a string of MCP broker hardening steps (GitHub token auth, tool allowlisting, remote spec validation), clearer private-IP guard rejections, and a new subcommand to load session briefs into Claude Code.

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

01 Managed config layer via CPLT_MANAGED_CONFIG

Not checked yet

NEW

80

cplt now supports a managed config layer driven by the `CPLT_MANAGED_CONFIG` environment variable, supplying organisation-level policy that sits above per-user settings and overrides a developer's local `~/.config/cplt/config.toml`.

Pin organisation-wide sandbox policy so every developer's cplt run inherits it, regardless of their local `~/.config/cplt/config.toml`.

```
$ export CPLT_MANAGED_CONFIG=/etc/cplt/managed.toml
cplt -- -p "fix the tests"
```

— Named env var plus a runnable export-and-run example 2026.11.24-020236-4ea5789

02 MCP broker authentication, tool restriction, and validation

Not checked yet NEW

75

cplt's MCP broker now serves OpenCode v2's GitHub MCP through a `gh-token` broker, letting OpenCode reach GitHub's MCP server without exposing raw credentials. The broker also supports GitHub token credentials for authenticating other remote MCP servers, validates remote specs before connecting to them, and lets operators restrict which MCP tools it exposes by listing them in `tools.allow` in the remote-leg configuration.

— Names broker config keys but no runnable example 2026.11.23-235509-ed1cfd0

2026.11.23-214351-f562118

THINNER COVERAGE BELOW

03 Session brief loading via brief subcommand

Not checked yet NEW

50

A new `brief` subcommand loads a session brief into Claude Code at sandbox startup.

— Named subcommand but no usage example shown 2026.11.23-210614-c0112b2

04 **Named reasons for private-IP guard rejections** Not checked yet

IMPROVED 30

The proxy now names the exact rule when the private-IP guard refuses a host, so practitioners can see which policy triggered the block.

— No example of the message format or rule names given 2026.11.23-220119-e3ce501

WAS THIS USEFUL? 👍 👎

◆ Applied AI Agents

Abridge

Feedback ⓘ announced · Nov 24, 2026 BLOG ↗ RANK

Abridge is an ambient AI that turns clinician-patient conversations into clinical notes.

> Abridge's November 2026 update adds a self-serve Impact Dashboard and user labeling to the Enterprise Portal, brings pre-admission and pre-round summaries to the Haiku mobile app, expands pre-visit summary data and persistence, and tailors notes for endocrinology.

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

01 **Impact Dashboard in Enterprise Portal** Not checked yet NEW

55

A new self-serve dashboard in the Enterprise Portal gives admins and reporting users visibility into active clinicians, note volume, usage by care setting and specialty, and adoption trends without needing to export data.

– Clear UI location and metrics named, no exact navigation path launch-20261124-f0748fb7

02 **User Labels and Tags for adoption reporting** Not checked yet NEW

55

Admins can add labels such as region, sub-division, and user type to user records in the Enterprise Portal, either one at a time or in bulk, then filter and group adoption reporting by those labels.

– Names the label types and bulk option, but no exact UI steps launch-20261124-f0748fb7

03 **Endocrinology-tailored note generation** Not checked yet NEW

55

Endocrinology notes are now tailored to the specialty, capturing insulin pump settings, CGM metrics, medication doses, relevant lab values, and pediatric growth and puberty evaluations in the HPI and Assessment & Plan sections.

– Specific clinical fields named, no setup or toggle described launch-20261124-f0748fb7

04 **Pre-Admission and Pre-Round Summaries in Haiku mobile app**

Not checked yet NEW

50

Pre-Admission and Pre-Round Summaries are now available in the Haiku mobile app for inpatient clinicians, turning on automatically for health systems already using these summaries in Hyperspace.

– Explains auto-activation condition but no manual setup steps launch-20261124-f0748fb7

05 **Pre-visit summaries: expanded data and persistent tab**

Not checked yet IMPROVED 45

Pre-visit summaries now include procedure, radiology, and pathology results alongside existing EHR data, and they now stay open as a tab alongside the clinical note and patient visit summary after a note is generated.

— Names new data sources and UI behavior change, no further mechanism

launch-20261124-f0748fb7

WAS THIS USEFUL? 👍 👎

Feynman

Feedback ⓘ v0.5.73 NOTES ↗ RANK

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

> Feynman's latest release improves visibility into concurrent and nested background subagent runs and adds automatic cleanup of old workflow data to prevent unbounded disk growth.

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

01 **Workflow retention cleanup after 30 days** Not checked yet IMPROVED 48

Finished workflows and their subagents are now deleted from disk after the 30-day retention window, preventing unbounded disk growth on long-running installs.

— Clear mechanism and number, but automatic with no user action v0.5.73

02 **Subagent identification and nested status visibility**

Not checked yet NEW 47

Concurrent runs of the same subagent can now be told apart using a short `label` shown in status output, and nested subagents started by a background subagent now appear in their parent's status view.

— Names the `label` field but no exact usage shown v0.5.73

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Serving & Optimization

NVIDIA Dynamo

Feedback ⓘ v1.7.0 NOTES ↗

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

Dynamo v1.7.0 adds cross-pool routing, deadline queues, batch scheduling, live planner retuning, speech-to-text and batch inference APIs, image editing, multinode Sidecar resilience, and simulation-backed Kubernetes deployments.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.7.0 https://github.com/ai-dynamo/dynamo.git
# already have the repo? check out this version:
$ git checkout v1.7.0
```

Enable cross-pool session affinity routing in Dynamo Frontend with `--router-policy affinity`, which selects a pool by load and pins each session to that pool Not checked yet

Update Dynamo Planner budgets, SLA targets, and scaling thresholds live without a restart via the `/engine/planner/config` endpoint Not checked yet

Update Dynamo Router routing parameters and busy thresholds live via the `/router_policy` endpoint Not checked yet

Dynamo Frontend adds an OpenAI-compatible `/v1/audio/transcriptions` endpoint for speech-to-text transcription served by a dedicated vLLM worker Not checked yet

Dynamo Frontend adds a Batch API accessible via `/v1/files` and `/v1/batches` Not checked yet

+16 more

UPDATED SUPPORTED PLATFORM Engine version updates

NEW SUPPORTED PLATFORM Muse Spark Open and Beam 1.0 model support

WAS THIS USEFUL? 👍 👎

◆ ML Platforms & SDKs

emisar

Feedback ⓘ v0.61.0 NOTES ↗ RANK

Emisar lets AI agents securely run infrastructure actions through MCP, with approval controls that protect production systems.

> Emisar's latest release adds a Sentry investigation pack for pulling stack traces and event details into runs, alongside smaller improvements across GCP diagnostics, Vault, key management, and credential validation.

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

01 Sentry issue event investigation pack

Not checked yet NEW

88

New `sentry.issue_latest_event` pack surfaces standalone, exception, and thread frames, plus message, log entry, contexts, tags, and breadcrumbs for an issue. Run output holds up to 1 MiB with continuation support when the preview is clipped.

Pull full Sentry event details — including stack frames, breadcrumbs, and tags — into a run to investigate an exception.

```
$ sentry.issue_latest_event
```

— Names exact pack, fields returned, and 1 MiB output limit. v0.61.0

THINNER COVERAGE BELOW

02 Vault list actions return empty collection

Not checked yet IMPROVED

45

Vault list actions now return an empty collection when there is nothing to list, instead of an error.

— Clear before/after behavior but no named endpoint. v0.61.0

03 Credential and signing key validation

Not checked yet IMPROVED

45

The runner now validates refreshed credentials before use, and the MCP bridge rejects unsupported signing keys at startup.

— States two validation behaviors without named config or flags. v0.61.0

04 Key creation expiry presets

Not checked yet IMPROVED

40

Key creation now offers expiry presets and shows the validated expiration date before you submit.

— Gives a clear UI starting point but few specifics. v0.61.0

05 **GCP Policy Troubleshooter read-only diagnostics** Not checked yet

IMPROVED

35

GCP Policy Troubleshooter now adds read-only access diagnostics.

— Names the feature area but no mechanism detail. v0.61.0

06 **Runs agent filter shows owner and key** Not checked yet IMPROVED

35

The Runs agent filter now shows the owner and key to distinguish agents that share the same name.

— Describes UI change with no further mechanism. v0.61.0

WAS THIS USEFUL? 👍 👎

Databricks AI

Feedback ⓘ announced · Nov 24, 2026 BLOG ↗

RANK

Databricks AI provides managed model serving, agent development, evaluation, and retrieval over data in the Databricks platform.

> Databricks AI ships a native Lakebase Postgres sink for Spark Structured Streaming, now generally available on Databricks Runtime 18 LTS and above, with external PostgreSQL support in Public Preview on DBR 19+.

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

01 **Lakebase Postgres sink for Structured Streaming** Not checked yet

NEW INTEGRATION NEW 91

A single `writeStream` call can now write computed streaming results directly to Lakebase using .toTable() for Unity Catalog-registered tables or format("postgresql") for unregistered tables or external PostgreSQL databases (e.g. Amazon RDS, self-hosted), the latter available in Public Preview on DBR 19+ via a Unity Catalog connection — replacing custom `ForeachWriter` implementations. Upsert semantics are declared with the `upsertkey` option so the sink maintains one row per key, with declarative merge strategies (DBR 19+) controlling how non-key columns update on conflict without hand-written `ON CONFLICT` SQL. Write throughput can be tuned with `batchinterval` and `batchsize` options, and the connector works with micro-batch or Real-Time Mode triggers; on serverless compute, `Trigger.AvailableNow` is used.

Write to an external PostgreSQL database (such as Amazon RDS) via a Unity Catalog connection when the target table is not registered in Unity Catalog.

```
$ df.writeStream
  .format("postgresql")
  .option("dbtable", "public.user_activity")
  .option("upsertkey", "user_id")
  .start("<lakebase_or_rds_endpoint>")
```

— Names every flag and includes a runnable code example launch-20261123-39bdded8

announced · Nov 24, 2026 BLOG ↗

Databricks adds Tag Exchange BETA to share Unity Catalog table- and column-level tags via the Apache Iceberg REST Catalog `labels` field for discovery, governance, and agent workflows.

Enable Tag Exchange via the Databricks Previews portal to expose Unity Catalog's table- and column-level tags through the `labels` field in the Apache Iceberg REST `LoadTable` response, making classifications consumable by any Iceberg REST-compatible platform, engine, or agent. Not checked yet

Unity Catalog now surfaces table- and column-level tags as string key-value pairs through the Apache Iceberg REST Catalog API's `labels` field, allowing downstream tools to use business context such as ownership, domain, and sensitivity without custom sync pipelines. Not checked yet

WAS THIS USEFUL? 👍 👎

Tracer Cloud OpenSRE

Feedback ⓘ v0.1.2026.11.24 NOTES ↗

RANK

OpenSRE is an open-source toolkit for building AI SRE agents.

> OpenSRE's hosted mode now defaults to GPT-6.2 Sol for LLM-backed agent turns.

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

01 **GPT-6.2 Sol as default hosted model** Not checked yet IMPROVED

30

OpenSRE's hosted mode now defaults to GPT-6.2 Sol for LLM-backed agent turns.

— Names the model but no mechanism or configuration detail v0.1.2026.11.24

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.58.0 NOTES ↗ RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> OpenAI Python SDK v3.58.0 adds the ability to set and read spending limits on agent sessions.

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

01 **Spending limits on agent sessions** Not checked yet NEW

37

The SDK's agent session API now supports setting and reading spending limits on agent sessions.

— No specific method names, params, or usage examples given v3.58.0

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.63.0 NOTES ↗ RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> OpenAI TypeScript SDK v7.63.0 adds the ability to set spending limits on agent sessions.

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

01 **Spending limits for agent sessions** Not checked yet NEW

35

Agent session parameters now support setting spending limits via the API.

— No exact param name or endpoint given v7.63.0

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore

Feedback ⓘ

Runtime: Instances available in AWS GovCloud (US-West) (2026-11-23) BLOG ↗

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

NEW SUPPORTED PLATFORM Amazon Bedrock AgentCore Runtime Instances in AWS GovCloud (US-West)

WAS THIS USEFUL? 👍 👎

Anthropic TypeScript SDK

Feedback ⓘ sdk-v1.13.0

NOTES ↗ RANK

The Anthropic TypeScript SDK provides typed JavaScript and TypeScript clients for Claude APIs, streaming, and tool use.

> The Anthropic TypeScript SDK added structured error reporting for Message Batches creation and introduced External Storage management with corresponding webhook notifications in the Admin API beta.

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

01 **External Storage management in Admin API beta** Not checked yet

NEW

43

Adds External Storage to the Admin API beta, enabling management of external storage configurations through the Admin API, plus new webhook events for external storage failure and recovery so applications can react programmatically when external storage becomes unavailable or comes back online.

— Describes capability and events but no endpoint names or payload sdk-v1.13.0

02 **Error code in Message Batches creation** Not checked yet NEW

37

The Message Batches creation response now includes an error code field, letting callers inspect structured failure reasons when a batch creation request is rejected.

— Names the field but no code example sdk-v1.13.0

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ v2.13.0 NOTES ↗ RANK

The Anthropic Python SDK provides typed Python clients for Claude APIs, streaming responses, and tool use.

> The Anthropic Python SDK's v2.13.0 release adds error visibility to Message Batches, introduces External Storage management in the Admin API beta, and adds webhook events for external storage lifecycle changes.

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

01 **Error code field on Message Batches creation** Not checked yet

IMPROVED 33

Message Batches creation responses now include an error code field, giving callers a machine-readable reason when batch creation fails.

— Names the field but not its exact key or possible values v2.13.0

02 **Webhook events for external storage failure and recovery**

Not checked yet NEW 33

New webhook events fire on external storage failure and recovery, letting applications react programmatically to storage lifecycle changes.

— Event existence described but no event names or payload v2.13.0

03 **External Storage support in Admin API beta** Not checked yet NEW

30

The Admin API beta gains External Storage support, enabling management of external storage resources through the SDK.

— No endpoint, method, or config key given v2.13.0

WAS THIS USEFUL? 👍 👎

◆ AI / LLM Gateways

BerriAI LiteLLM

Feedback ⓘ CVE-2026-93355 CVE ↗ STABLE

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

`CVE-2026-93355`: JWT and SSO email fallback can sign in as another user when the IdP lets users choose their email address (CVSS 8.1, affects `<1.110.0`)

Not checked yet

NEW INTEGRATION Gradium text-to-speech

NEW INTEGRATION Gnani native Prisma v2.5 and v3 REST transcription

NEW INTEGRATION Perplexity Orchestrator provider with cost tracking

NEW SUPPORTED PLATFORM gemini-decision-1 on /v1/decisions and /v1/systemone

▾ 1 more release in this issue · 2026-11-23 → 2026-11-24

v1.108.16 NOTES ↗ LATEST STABLE

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/berriai/litellm:v1.108.16
```

git

```
$ git clone --branch v1.108.16
https://github.com/BerriAI/litellm.git
# already have the repo? check out this version:
$ git checkout v1.108.16
```

published under the project's GitHub account · checked 24 Nov Wrong command?

WAS THIS USEFUL? 👍 👎

Mozilla AI Otari

Feedback ⓘ v0.55.0 NOTES ↗ RANK

Mozilla AI Otari is an open-source AI gateway you run yourself. One endpoint for 40+ providers, with virtual keys, budgets, and usage tracking.

> Otari's v0.55.0 release sharpens guardrail observability with category-level reporting and a stable error code, adds a discoverable `temperature` capability flag to its model catalog, and changes the identity record schema for proved addresses.

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

01 **`temperature` field in catalog `supported_parameters`**

Not checked yet NEW 65

Catalog entries now list `temperature` in `supported_parameters`, so callers can discover whether a model accepts temperature before sending it.

– Names exact field, lacks endpoint or example call v0.55.0

THINNER COVERAGE BELOW

02 **Stable error code for guardrail 502 responses** Not checked yet

IMPROVED 50

Guardrail 502 responses now carry a stable error code, making programmatic handling of guardrail failures reliable.

– Names the status code but not the code value itself v0.55.0

03 **Identity record schema change for proved addresses**

Not checked yet 50

BREAKING

Identity records now store that an address was proved (not that it was not proved) — existing identity records may need migration to match the new schema.

– Flags migration need but no migration steps given v0.55.0

04 **Category detail in guardrail blocks and traces** Not checked yet

NEW 45

Guardrail blocks now report which categories fired, giving responders specific signal about what triggered a block, and traces record the same category-fired detail so post-incident review has category-level information.

– Describes behaviour but no API shape or example v0.55.0

└──▷ **BREAKING ON UPGRADE**

! Identity records now store that an address was proved (not that it was not proved) — existing identity records may need migration to match the new schema.

WAS THIS USEFUL? 👍 👎

◆ Inference Providers & GPU Clouds

AWS Amazon Bedrock

Feedback ⓘ

New topic: Trusted access programs in Amazon Bedrock (2026-11-23) BLOG ↗

RANK

Amazon Bedrock provides managed foundation-model inference, customization, knowledge bases, and APIs for generative AI applications.

> Amazon Bedrock introduced trusted access programs for managing model-provider offers that change model availability or request handling, and added a case-handling workflow under Enterprise Frontier Safeguards and Enterprise Data Controls for Anthropic models in public preview.

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

01 Trusted access programs for model providers

Not checked yet NEW

80

Amazon Bedrock now supports trusted access programs — offers from model providers that change which models are available or how requests are handled — with controls to opt a project, account, or organization in or out, enforced via IAM and service control policies. In the console, users navigate to Trusted access programs to check enrollment status and request approval, and admins can attach a service control policy in AWS Organizations to deny the trusted-access enrollment action org-wide.

Manage trusted access program enrollment for a project, account, or organization using IAM and service control policies.

📍 In the Amazon Bedrock console, navigate to Trusted access programs, find the program offered by your model provider, check your enrollment status, and request approval. To restrict enrollment org-wide, attach a service control policy in AWS Organizations that denies the trusted-access enrollment action.

— Names enforcement mechanism and exact console path, no API/flag given

New topic: Trusted access programs in A…

THINNER COVERAGE BELOW

02 Case-handling workflow for Anthropic EFS/EDC

Not checked yet

NEW

35

Amazon Bedrock adds a case-handling workflow under Enterprise Frontier Safeguards (EFS) and Enterprise Data Controls (EDC) for managing retained content for Anthropic models, currently in public preview.

— Names the feature area but no concrete steps or surfaces

Updated topic: EFS and EDC for Anthropi…

WAS THIS USEFUL? 👍 👎

Nebius AI Cloud

Feedback ⓘ November 16–22, 2026 (2026-11-23)

BLOG ↗

Nebius AI Cloud provides GPU compute, managed Kubernetes, distributed training infrastructure, and serverless AI workloads.

Nebius AI Cloud adds load-balancer security-group restrictions for Managed Kubernetes Services via annotation or Service specification update.

NEW INTEGRATION Restrict access to a Managed Kubernetes load balancer by assigning a security group to the Service through an annotation or by updating the Service specification. Not checked yet

NEW SUPPORTED PLATFORM eu-north2 region (Iceland)

WAS THIS USEFUL? 👍 👎

Anyscale

Feedback ⓘ Version 0.26.119 release notes NOTES ↗ RANK

Anyscale provides a managed Ray platform for distributed AI development, training, batch processing, and model serving.

> Anyscale added a `--skip-install` flag to `anyscale cloud setup --stack kuberay` for GitOps-managed Connector installs.

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

01 **GitOps-friendly KubeRay Connector setup flag** Not checked yet

NEW 90

Passing `--skip-install` to `anyscale cloud setup --stack kuberay` skips Connector chart installation and instead writes Helm values to `anyscale-connector-values.yaml` (or a custom path via `--values-file`); the command still creates the S3 bucket and IAM role, registers the cloud, and prints a `helm install` command and `kubectl` command for finishing the install manually.

Register a KubeRay cloud and emit Helm values for a GitOps pipeline to install the Connector, rather than having the CLI install it directly.

```
$ anyscale cloud setup --stack kuberay --skip-install --name my-kuberay-cloud --values-file ./infra/anyscale-connector-values.yaml
```

— Names exact flags, output file, and follow-up commands. Version 0.26.119 release notes

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

vMLX

Feedback ⓘ v1.6.85 NOTES ↗ RANK

The vMLX server runs compressed MLX models on Apple Silicon with disk caching, paged memory, continuous batching, and hybrid SSM scheduling.

> vMLX's v1.6.85 release adds full support for the Inkling model family — multimodal routing, native weight loading, DSpark speculative decoding, independent cache lanes, and SSD draft checkpoints — alongside improvements to Electron proof capture.

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

01 Native MLX loader support for Inkling weight formats

Not checked yet NEW 60

The native MLX loader can now load Inkling Mini native packed weights, JANGH2 expert and loop weights, and BF16 DSpark weights, covering multiple bundle layouts.

– Names weight formats but not the loader invocation v1.6.85

THINNER COVERAGE BELOW

02 SSD draft checkpoint support for Inkling Not checked yet NEW

56

Inkling SSD draft checkpoints are now supported, with prefix-cache clearing, sliding-window normalization, and snapshot restore.

– Names three mechanisms but no config or command shown v1.6.85

03 Inkling DSpark speculative decoding in multimodal scheduler

Not checked yet NEW 52

Inkling DSpark speculative decoding now runs through the normal multimodal scheduler, with request-scoped verification stats exposed.

– Describes integration but no way to inspect the stats shown v1.6.85

04 Multimodal routing and tool calls for Inkling Mini

Not checked yet NEW 46

Inkling Mini images and native tool calls can now be routed through the app and API, enabling multimodal and agentic use of the Inkling model family.

– Names routing surface but no flag or endpoint given v1.6.85

05 Independent attention and convolution cache lanes for Inkling

Not checked yet NEW 44

Inkling attention and convolution cache lanes are now exposed independently, enabling fine-grained cache control.

— Names the lanes but not how to control them v1.6.85

06 **Reasoning separation and selector verification in Electron proof**

Not checked yet IMPROVED 44

Electron proof capture now separates answer and reasoning, with per-reasoning-mode settings validation, and verifies the native LiLiCorr selector in the proof.

— Describes two related proof-capture changes without usage detail v1.6.85

WAS THIS USEFUL? 👍 👎

llmfit

Feedback ⓘ v1.1.21 NOTES ↗ CODE ↗ RANK

Llmfit hundreds of models & providers. One command to find what runs on your hardware.

> llmfit's v1.1.21 release adds a tool-call probing mode to its bench command and brings German localization to the dashboard and desktop app.

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

01 **Tool-call support probing in bench** Not checked yet NEW

75

The `--tool-calls` flag on `llmfit bench` probes whether a model correctly returns OpenAI-style tool calls, letting users verify support before wiring a model into an agentic pipeline.

Verify that a specific model correctly returns tool calls before wiring it into an agentic pipeline.

```
$ llmfit bench --tool-calls
```

— Names exact flag and command, with a runnable example v1.1.21

THINNER COVERAGE BELOW

02 **German translations added** Not checked yet NEW

25

llmfit adds German translations for the web dashboard, desktop app, and README.

— Brief, no mechanism or specifics beyond surfaces listed v1.1.21

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Tencent WeKnora

Feedback ⓘ v0.9.1 NOTES ↗ RANK

WeKnora is an open-source LLM knowledge platform: turn raw documents into a queryable RAG, an autonomous reasoning agent, and a self-maintaining Wiki.

> WeKnora's v0.9.1 release adds a raft of integrations — new data source connectors, web search and model providers — alongside sandbox session isolation, voice input, OIDC PKCE S256 auth, and a cluster of knowledge-base and wiki improvements.

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

01 Postgres connection pool cap via env var

Not checked yet NEW

85

The `DB_MAX_OPEN_CONNS` environment variable caps the Postgres connection pool size (example: `DB_MAX_OPEN_CONNS=25` in `.env`) to prevent the pool from exhausting database connection slots on resource-constrained hosts.

Cap Postgres connections on a resource-constrained host to prevent the pool from exhausting database slots.

```ini
DB_MAX_OPEN_CONNS=25
```

— Exact env var, file and example value given v0.9.1

02 New data source connectors and model/search providers

Not checked yet NEW

65

Adds a Nextcloud `WebDAV` connector as a data source for automatic knowledge-base sync and syncs document attachments from Confluence, plus `DeepSeek` as a web search and fetch provider, `SearchApi` as a web search provider, Runcrate as an OpenAI-compatible model provider, and Claude 5.5 models to the OpenRouter catalog.

— Names both connectors, no setup steps Names all four providers, no config steps v0.9.1

03 Expanded knowledge base and wiki capabilities

Not checked yet

NEW

60

Adds image-based knowledge search, editing of image OCR text and captions in knowledge chunks, manual-only knowledge bases in the wiki, browsing knowledge folders when mentioning a file in chat, and selecting an independent question-generation model for knowledge bases.

— Names five distinct additions without usage steps v0.9.1

THINNER COVERAGE BELOW

04 **OIDC PKCE S256 authentication support** Not checked yet NEW

40

Adds opt-in support for OIDC PKCE S256 for stronger authorization-code authentication flows.

— Names the flow but no config steps given v0.9.1

05 **Hold-to-talk voice input for agent** Not checked yet NEW

40

Adds hold-to-talk voice input backed by the agent's ASR model.

— Names ASR backing, no UI path given v0.9.1

06 **Session-isolated sandbox desktops** Not checked yet NEW

35

Adds session-isolated desktops and agent-browser login sharing in the sandbox.

— Brief description, no mechanism detail v0.9.1

07 **Browser-local chat history in embed widget** Not checked yet NEW

30

Adds browser-local chat history for the embed widget.

— Brief, no storage mechanism detail v0.9.1

08 **Tenant-admin subject management for memory** Not checked yet NEW

25

Adds tenant-admin subject management for memory.

— Single-line description, no mechanism given v0.9.1

WAS THIS USEFUL? 👍 👎

GrowGraph OntoCast

Feedback ⓘ v0.9.3 CODE ↗ RANK

OntoCast generates RDF knowledge graphs from documents by co-evolving domain ontologies and fact graphs with an agentic LLM pipeline.

> OntoCast's v0.9.3 release adds provenance-replay output for its facts pipeline, expands fact-checking with several new validation checks, and ships a round of breaking API renames across the statement, scope, and quarantine contracts.

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

01 Provenance replay via `.units.trig` snapshots

Not checked yet

NEW

85

Running `ontocast process --keep-provenance` now writes intermediate unit facts to a `<name>.units.trig` file, so aggregation can be replayed from that saved snapshot without issuing a new model call.

Replay aggregation on previously processed units without issuing new model calls, by running with `--keep-provenance` and reusing the saved `.units.trig` snapshot.

```
$ ontocast process --keep-provenance my_document.txt
```

▶ View command output v0.9.3

— Names exact flag, output file, and runnable command v0.9.3

02 Expanded fact-checking validations and findings

Not checked yet

NEW

85

`check_facts` now validates every number in a value against the unit's text (digits or matching number words) via the new `text2num` dependency, flagging `ungrounded_number` and `contradicted_individual`; new `twin_labels` and `facts_twin_labels` checks compare labels only between unlinked instances of the same role (statements, observed properties, other instances), leaving values out; aggregation keeps two entities apart when their statements report one property with conflicting results or units of different dimensions, surfacing `statement_conflict` and `dimension_conflict`; a new `role_outside_statement` finding flags a statement's result, subject, property, or context link held by a node whose declared side does not admit it, removing the link at exit; and a new `empty_value` finding flags and removes a value node carrying nothing but its types and annotations at exit.

— Names every finding and check but no invocation path v0.9.3

03 Facts pipeline API renames

Not checked yet

80

BREAKING

`StatementContract` replaces `SubjectContract`, `WellKnownVocabulary.statement_roles` replaces `statement_properties` and `subject_links`, and AtomicToolBox.statement_contract() replaces subject_contract(). `FactScope` replaces `fact_instances`, `is_scaffolding`, and `without_scaffolding`. `write_back_quarantine` replaces `write_back_temporal_quarantine`; `RejectedLiteralTriple.expected_datatypes` replaces `expected_datatype`. `missing_role` replaces `statement_without_subject`; telemetry keys `facts_statements_missing_role*` and `n_roles_filled` replace previous names. `AggregationResult.unconfirmed_merges` is removed.

— Full rename mapping gives a clear migration path v0.9.3

---

04 **attributed_to role for cited statements** Not checked yet NEW

65

Facts prompts now ask that a statement attributed to other work link to it via the new `attributed_to` role of `CHUNK_CITATION_VOCABULARY` (default `dcterms:source`).

— Names vocabulary role and default but no usage example v0.9.3

---

05 **Cap on LLM coreference candidate pairs** Not checked yet NEW

65

LLM coreference candidate pairs can now be capped with the new `AGG_LLM_COREFERENCE_MAX_PAIRS` environment variable, which defaults to 80.

— Names exact env var and default value v0.9.3

---

06 **Docker containers bind to localhost only** Not checked yet BREAKING

60

The Fuseki and Qdrant containers under `docker/` now publish their ports on `127.0.0.1` only; recreate a running container to apply the change.

— Names affected containers and the required remediation step v0.9.3

THINNER COVERAGE BELOW

---

07 **QUDT currency units and term suggestions in catalog**

Not checked yet NEW

55

The term catalog now ships QUDT currency units, and a non-catalog term is offered catalog terms of its kind whose local name matches after folding case and separators; suggestions are never applied automatically.

— Describes mechanism but no config or command to try it v0.9.3

---

08 **Ontology planning mints missing well-known terms** Not checked yet

IMPROVED

55

Ontology planning now mints a term in the home ontology when a well-known namespace lacks it (for example, `schema:leader`), instead of reusing an absent term.

— Clear before/after but no config exposed v0.9.3

└──▷ **BREAKING ON UPGRADE**

! `StatementContract` replaces `SubjectContract`, `WellKnownVocabulary.statement_roles` replaces `statement_properties` and `subject_links`, and AtomicToolBox.statement_contract() replaces subject_contract().

! `FactScope` replaces `fact_instances`, `is_scaffolding`, and `without_scaffolding`.

! `write_back_quarantine` replaces `write_back_temporal_quarantine`; `RejectedLiteralTriple.expected_datatypes` replaces `expected_datatype`.

! `missing_role` replaces `statement_without_subject`; telemetry keys `facts_statements_missing_role*` and `n_roles_filled` replace previous names.

! `AggregationResult.unconfirmed_merges` is removed.

! The Fuseki and Qdrant containers under `docker/` now publish their ports on `127.0.0.1` only; recreate a running container to apply the change.

WAS THIS USEFUL? 👍 👎

lightrag

Feedback ⓘ v1.5.8 NOTES ↗ RANK

LightRAG is a lightweight knowledge-graph RAG framework combining dual-layer graph and vector retrieval for efficient document question answering.

> LightRAG's v1.5.8 release centers on configuration-storage safety — a selectable backend, an anchor file binding deployments to their backend, and embedding-baseline enforcement — plus four new offline maintenance CLI tools, alongside a long list of breaking changes touching Python version, authentication, chunk-size validation, query error handling, and API endpoints.

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

01 New offline maintenance CLI tools

Not checked yet NEW

90

Four new offline tools ship alongside the server: `lightrag-migrate-config` moves configuration between backend types offline (e.g. `lightrag-migrate-config --target-backend PGKVStorage` to move from JSON files to PostgreSQL), `lightrag-clear-storage` clears a workspace offline including configuration storage and the anchor file, `lightrag-repair-chunk-tracking` removes orphan chunk-tracking rows left by interrupted document deletions, and `lightrag-gc-llm-cache` reports or deletes stale LLM extraction cache rows (`--apply` to delete).

Move configuration storage from JSON files to PostgreSQL before switching to a database-backed deployment.

```
$ lightrag-migrate-config --target-backend PGKVStorage
```

Find and delete stale LLM extraction cache rows left behind by documents deleted in earlier versions.

```
$ lightrag-gc-llm-cache --apply
```

— Four named commands with runnable examples and flags v1.5.8

02 Anchor file binds deployments to configuration backend

Not checked yet

88

BREAKING

A new anchor file `WORKING_DIR/config_storage_anchor.json` binds a deployment to its configuration backend and container UUID on first start, so a changed `LIGHTRAG_CONFIG_STORAGE`, different database, or replaced volume is refused rather

than silently moving configuration. As a direct consequence, changing `LIGHTRAG_KV_STORAGE` after the first v1.5.8 start is now refused while `LIGHTRAG_CONFIG_STORAGE` is unset — fix by setting `LIGHTRAG_CONFIG_STORAGE` explicitly to the anchored backend, or by moving configuration first with `lightrag-migrate-config`.

Pin configuration storage to PostgreSQL so that changing `LIGHTRAG_KV_STORAGE` later does not silently move the configuration.

```ini
LIGHTRAG_CONFIG_STORAGE=PGKVStorage
LIGHTRAG_KV_STORAGE=PGKVStorage
```

— Names anchor file path, exact env vars and remediation path v1.5.8

---

03 **Configurable backend for configuration storage** Not checked yet

NEW

83

A new `LIGHTRAG_CONFIG_STORAGE` setting selects where configuration is stored, accepting `JsonKVStorage`, `PGKVStorage`, `MongoKVStorage`, or `OpenSearchKVStorage`; when unset it follows `LIGHTRAG_KV_STORAGE`, so most deployments need no new setting, but Redis deployments fall back to `JsonKVStorage` with a startup warning.

— Names env var, accepted values and fallback behavior precisely v1.5.8

---

04 **Ollama API routes require authentication** Not checked yet

BREAKING

83

Ollama `/api/*` routes now return HTTP 403 when `LIGHTRAG_API_KEY` is set and `WHITELIST_PATHS` is unset and no key is provided; in Open WebUI the `lightrag:latest` model disappears until the key is entered as the Ollama connection API key or `WHITELIST_PATHS=/health,/api/*` is set.

— Exact status code, env vars and Open WebUI remediation steps v1.5.8

---

05 **Configurable JSON mode for keyword extraction** Not checked yet

NEW

80

A new `KEYWORD_EXTRACTION_JSON_MODE` setting (`auto`, `json_object`, or `none`; default `auto`) controls query keyword extraction JSON mode and retries without JSON mode when a provider rejects it, fixing empty results on servers such as LM Studio.

— Exact setting name, values and the bug it fixes v1.5.8

---

06 **WebUI drops API-key authentication** Not checked yet

BREAKING

78

The WebUI (`/webui` and `/workspace`) no longer accepts or stores an API key; a server protected only by `LIGHTRAG_API_KEY` now shows a configuration guidance page — fix by setting `AUTH_ACCOUNTS` and a strong `TOKEN_SECRET` .

— Names affected paths and exact remediation settings v1.5.8

---

07 **Minimum 64-token chunk size enforced** Not checked yet BREAKING 75

`CHUNK_SIZE` , `CHUNK_F_SIZE` , `CHUNK_R_SIZE` , `CHUNK_V_SIZE` , `CHUNK_P_SIZE` , `addon_params` , `LIGHTRAG_PARSER` rules, and REST `chunking.params` now refuse values below 64 tokens, causing startup failure or HTTP 422.

— Names every affected config key and the new validation floor v1.5.8

---

08 **Milvus load timeout setting** Not checked yet NEW 70

A new `MILVUS_LOAD_TIMEOUT` setting (default 600 seconds) bounds how long Milvus collection loading waits at startup, preventing startup failures on large collections.

— Named config key with default and purpose v1.5.8

---

09 **Query functions raise exceptions on failure** Not checked yet 65

BREAKING

aquery(), query(), aquery_llm(), and query_llm() now raise exceptions on failure instead of returning None or `status == 'failure'` ; the API server now answers HTTP 500 on failure.

— Names affected functions and the new HTTP behavior v1.5.8

---

10 **Clear-cache endpoint replaced** Not checked yet BREAKING 65

`POST /documents/clear_cache` is removed; use `DELETE /documents?clear_llm_cache=true` instead.

— Exact old and new endpoint with method and path v1.5.8

---

11 **PGTableGraphStorage option in setup wizard** Not checked yet NEW 60

The setup wizard now offers `PGTableGraphStorage` (plain PostgreSQL tables, no Apache AGE) as a graph storage option and generates configuration from the official `pgvector/pgvector` image.

— Names the storage option and base image but no exact flags v1.5.8

THINNER COVERAGE BELOW

12 **Workspace names starting with $ reserved** Not checked yet BREAKING

55

Workspace names starting with `$` are now reserved; rename e.g. `WORKSPACE=$team` to `WORKSPACE=_team` (the name under which the data is already stored).

— Gives exact migration example for affected deployments v1.5.8

13 **Embedding baseline enforcement at startup** Not checked yet NEW

53

LightRAG now records each workspace's embedding baselines in configuration storage and refuses startup when the configured embedding model no longer matches stored vectors, preventing silent embedding-space mismatches.

— Mechanism described but no config key or remediation given v1.5.8

14 **Python 3.11 now required** Not checked yet BREAKING

35

Python 3.10 is no longer supported; Python 3.11 or later is now required to run LightRAG.

— States the requirement with no further detail v1.5.8

└──▷ **BREAKING ON UPGRADE**

! Python 3.10 is no longer supported; Python 3.11 or later is required.

! Changing `LIGHTRAG_KV_STORAGE` after the first v1.5.8 start is refused while `LIGHTRAG_CONFIG_STORAGE` is unset — either set `LIGHTRAG_CONFIG_STORAGE` explicitly to the anchored backend, or move configuration first with `lightrag-migrate-config`.

! Ollama `/api/*` routes now return HTTP 403 when `LIGHTRAG_API_KEY` is set and `WHITELIST_PATHS` is unset and no key is provided; in Open WebUI the `lightrag:latest` model disappears until the key is entered as the Ollama connection API key or `WHITELIST_PATHS=/health,/api/*` is set.

! The WebUI (`/webui` and `/workspace`) no longer accepts or stores an API key; a server protected by `LIGHTRAG_API_KEY` alone now shows a configuration guidance page — fix by setting `AUTH_ACCOUNTS` and a strong `TOKEN_SECRET`.

! `CHUNK_SIZE`, `CHUNK_F_SIZE`, `CHUNK_R_SIZE`, `CHUNK_V_SIZE`, `CHUNK_P_SIZE`, `addon_params`, `LIGHTRAG_PARSER` rules, and REST `chunking.params` now refuse values below 64 tokens (startup failure or HTTP 422).

! aquery(), query(), aquery_llm(), and query_llm() now raise exceptions on failure instead of returning `None` or `status == 'failure'`; the API server answers HTTP 500 on failure.

! `POST /documents/clear_cache` is removed; use `DELETE /documents?clear_llm_cache=true` instead.

! Workspace names starting with `$` are reserved; rename `WORKSPACE=$team` to `WORKSPACE=_team` (the name data is already stored under).

WAS THIS USEFUL?  

Weaviate

Feedback ⓘ v1.41.0 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's v1.41.0 release and docs updates introduce licensed Enterprise and Platform editions with namespace isolation for multi-tenant deployments, dramatic runtime property reindex performance gains, a shard rebalancing preview, DB user API key TTLs, task filtering, HFresh multi-vector support, and union-type image generation in the Query Agent.

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

01 Filter distributed tasks by collection and tenant

Not checked yet

IMPROVED

80

Filter `GET /v1/tasks` by `collection` and `tenant` query parameters to narrow distributed-task queries, so callers can poll only the reindex tasks affecting a specific collection and tenant, e.g. `GET /v1/tasks?collection=MyCollection&tenant=tenant-42`.

Poll only the reindex tasks affecting a specific collection and tenant, so CI jobs can wait on exactly the work they triggered.

```
$ curl -X GET 'https://<weaviate-host>/v1/tasks?
collection=MyCollection&tenant=tenant-42' -H 'Authorization:
Bearer <token>'
```

— Exact endpoint, params and runnable curl example v1.41.0

02 Union types for conditional image generation in Query Agent

Not checked yet NEW

70

Make an image field nullable in a `UnionAsToolCallResponse` output type to let the Query Agent decide whether to generate an image based on the query, using the field description to guide the decision — an alternative to tool calls for conditional image generation in structured responses.

— Names exact type but no runnable example product docs

03 Faster runtime property reindex with partial resubmission

Not checked yet IMPROVED

70

Runtime property reindex now supports resubmitting a partial reindex without forcing finished tenants back to HOT state. With runtime reindex enabled, a node with 100k tenants now starts in 6 seconds instead of 16 minutes, and reindex shard lookups at 1 million tenants complete in approximately 0.02 seconds per phase instead of approximately 45 hours.

— Concrete before/after numbers but no command to trigger it v1.41.0

04 **Namespace isolation for multi-tenant deployments** Not checked yet

NEW

65

The Platform Edition adds namespaces — isolated virtual clusters, one per team or end user, managed centrally as a shared service. Namespaces support replication factor 3, can run across multiple home nodes, let operators set custom suspend messages, enforce cross-namespace alias prevention so operator aliases cannot cross namespace boundaries, and enforce a per-namespace object cap on replicated writes.

— Names several namespace controls but no config keys v1.41.0

05 **License-key activation for Enterprise and Platform Editions**

Not checked yet NEW

63

Activate the Enterprise Edition or Platform Edition by setting a license key environment variable and restarting Weaviate — all editions ship in the same Docker image and binary, so no new image or download is needed. Each licensed feature has its own environment variable that turns it on; if enabled without a valid license key, Weaviate logs a warning and declines requests at runtime rather than failing to start.

— Mechanism explained but exact env var names not given product docs

THINNER COVERAGE BELOW

06 **Semi-automatic shard rebalancing (preview)** Not checked yet NEW

45

Semi-automatic shard rebalancing, shipped as a preview, adds REST endpoints for the full rebalance-plan journey plus a rebalance measurement endpoint and client.

— Endpoints mentioned but not named v1.41.0

07 **HFresh support for multi-vector vectorizers** Not checked yet NEW

40

Allow HFresh with multi-vector vectorizers, with MUVERA implied and required for multi-vector HFresh.

— Names HFresh and MUVERA but no usage steps v1.41.0

08 **Time-to-live for database user API keys** Not checked yet NEW

30

Add a time-to-live to DB users so API keys expire automatically.

— No config key or expiry mechanism named v1.41.0

09 **Async replication allowed during node maintenance** Not checked yet

IMPROVED

25

Allow async replication while a node is in maintenance mode.

— Single line with no mechanism detail v1.41.0

WAS THIS USEFUL? 👍 👎

ThHanke Ontosphere

Feedback ⓘ v.1.8.2 NOTES ↗ RANK

Ontosphere is a browser-based RDF/ontology knowledge graph editor — load RDF from files, URLs or SPARQL endpoints; author nodes and edges on the canvas; run OWL-RL reasoning; apply multi-algorithm layout and clustering. AI-native via MCP tool surface and relay bridge.

> Ontosphere's v1.8.2 release sharpens its reasoning and validation surfaces — SHACL now reports per-shape focus counts and comparisons, entailment checks return a decisive verdict, and the reasoner handles SWRL rules, ill-typed literals and transitive imports more faithfully — alongside a new JSON-LD dataset export format.

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

01 **SHACL focus counts and comparison in validation** Not checked yet

NEW 90

`validateGraph` now returns a `focus` object (`nodeShapes` , `shapesWithFocus` , `perShape`) and, after running with and without inferred triples, a `focusComparison` field, letting practitioners distinguish a shape that selects nothing from one that selects nodes and conforms. `runReasoning` also returns `shaclFocus` and `shaclFocusComparison` alongside the existing reasoning report.

Check whether a shape file is silently excluding all nodes — open the SHACL panel after loading shapes to see per-shape focus counts and spot shapes with zero focus.

📍 In the SHACL panel, review the focus counts line in the reasoning report; a shape showing 0 under shapesWithFocus selects no nodes and is now distinguishable from a shape that selects nodes and passes.

— Names exact returned fields and gives a UI check procedure v.1.8.2

02 **reasonerClosure setting for full subClassOf closure**

Not checked yet NEW 88

A new `reasonerClosure` setting in the app config writes the full transitive `rdfs:subClassOf` class hierarchy into the inferred graph; it is Konclude-backend only and off by default.

Turn on full class-hierarchy materialisation when SPARQL or SHACL queries need to traverse the complete `rdfs:subClassOf` closure without an active reasoner.

```yaml
reasonerClosure: true
```

– Exact config key with example value and backend scope v.1.8.2

03 **Verdict field in explainEntailment** Not checked yet IMPROVED

78

`explainEntailment` now returns a `verdict` field of `entailed`, `not-entailed`, or `undetermined`, populating `undeterminedKind` and `reason` when the check cannot decide; undecided answers are not cached.

– Names exact field values but no usage walkthrough given v.1.8.2

04 **Failed SHACL shape file indicators** Not checked yet NEW

71

Failed SHACL shape files now surface in the SHACL panel with a red N failed badge on the SHACL Shapes header, a dot on the collapsed sidebar button, and an alert listing each file with its error; the `loadShacl` MCP tool reports failures through the same track.

– Describes exact UI locations but no runnable command v.1.8.2

05 **Ill-typed literal warnings in reasoning report** Not checked yet

NEW

71

Ill-typed literals (e.g. `"5abc"^^xsd:integer`, `"300"^^xsd:byte`, integers beyond 64 bits) now appear as `reasoner:ill-typed-literal` warnings in the reasoning report, linked to the subject, so practitioners know when the reasoner silently drops input.

– Concrete warning code and examples but no direct action step v.1.8.2

06 **DL-safe SWRL rule execution during reasoning** Not checked yet

NEW

68

DL-safe SWRL rules (`swrl:Imp`) found in the data are now applied during reasoning by default, with their consequences appearing as inferred triples; rules using `swrlb:*` built-ins are skipped.

– Names rule types and default behaviour, no config/flag shown v.1.8.2

07 **jsonld-dataset export format** Not checked yet NEW

65

`exportGraph` gains a new `jsonld-dataset` format that writes one JSON-LD graph object per named graph, allowing named graphs to be exported individually.

– Names function and format string but no example call shown v.1.8.2

THINNER COVERAGE BELOW

08 **Transitive owl:imports resolution** Not checked yet IMPROVED

48

`owl:imports` are now followed transitively with a visited set and limits, preventing silent incomplete imports.

— Mechanism named but limits and visited-set details unspecified v.1.8.2

09 **SPARQL updates journaled as agent edits** Not checked yet NEW

40

SPARQL updates run through `queryGraph` are now journaled as agent edits.

— Bare statement, no mechanism or journal format detail v.1.8.2

WAS THIS USEFUL? 👍 👎

knowhere

Feedback ⓘ v1.2.68 NOTES ↗ RANK

Knowhere extracts, parses, and outputs structured chunks ready for AI Agents and RAG.

> Knowhere's latest releases add document usage tracking and importance-weighted retrieval ranking, alongside reliability improvements for embedding service failures, new Logfire tracing for retrieval routing, and tighter error codes across webhook and API key endpoints.

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

01 Usage recording for retrieval ranking

Not checked yet NEW

95

Call `POST /api/v2/retrieval/usage` with `namespace` and `document_ids` to record which documents an Agent actually used; only active documents the caller owns in that namespace are recorded. Usage recording can also be enabled on `/tools/invoke` calls by setting `records_usage` on the request, with hits counted only when the call succeeds. Automatic recall requests (`purpose=automatic`) no longer record hits — only explicit searches do — and at most one use per document or chunk is counted per UTC day.

Signal to knowhere which documents your agent actually consumed at the end of a retrieval turn, so their importance scores rise over time.

```
$ curl -X POST https://<host>/api/v2/retrieval/usage \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"namespace": "<namespace>", "document_ids": ["<doc-id-1>",
"<doc-id-2>"]}'
```

— Full endpoint, field, and constraints with runnable example. v1.2.68

02 Importance-weighted retrieval scoring

Not checked yet NEW

65

Retrieval scores are now weighted by document importance — a multiplier between 0.7 and 1.5 derived from usage heat and upload freshness, both halving every 30 days.

— Clear mechanism and numeric scaling but no user action. v1.2.68

03 Retryable 503 on embedding service unavailability

Not checked yet

NEW

60

Retrieval now returns a retryable `503` status when the required embedding service stays unavailable after retries, letting callers distinguish transient failures from hard

errors.

— Names exact status code, lacks retry count or backoff detail. v1.2.67

04 **Precise error codes for Stripe webhooks and API key toggling**

Not checked yet IMPROVED 60

Invalid Stripe webhook signatures and malformed payloads now return `400 INVALID_ARGUMENT` instead of a silent or ambiguous error, and API key toggling now returns `404 NOT_FOUND` when the key is unavailable to the current user.

— Exact status codes named, no endpoint paths given. v1.2.67

THINNER COVERAGE BELOW

05 **Logfire tracing for retrieval routing and fallback**

Not checked yet NEW 55

Logfire now records the automatic routing trigger, model configuration source, failure stage, and Agent fallback outcome for retrieval observability.

— Names tracked fields but no access path or query example. v1.2.67

06 **Hipo hit stats reset during ranking migration** Not checked yet

30

BREAKING

The migration tied to the new importance-weighted ranking resets existing Hipo hit stats, which were counted under the old rules.

— States impact but gives no migration steps. v1.2.68

07 **Corpus retrieval handles empty revision sets** Not checked yet

IMPROVED 30

Corpus retrieval now handles empty revision sets and returns connected assets in recall results.

— Thin description, no mechanism or endpoint named. v1.2.67

└──▷ **BREAKING ON UPGRADE**

! The migration resets existing Hipo hit stats, which were counted under the old rules.

WAS THIS USEFUL? 👍 👎

oxideterm

Feedback ⓘ v2.4.1 NOTES ↗ RANK

Oxideterm is a fast, native Rust + GPUI workspace for local and SSH with MCP tools—no WebView, no telemetry, open source.

> OxideTerm v2.4.1 adds multi-select batch session removal with automatic retention cleanup, visibility into AI prompt-cache hit rates, and faithful replay of multi-turn AI tool calls, alongside smaller terminal, connection, and remote-desktop improvements.

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

01 **Batch session removal and retention policy** Not checked yet NEW

80

The Active Sessions view and session manager sidebar now support multi-select batch removal via click, Ctrl-toggle, and Shift-range selection; removing a selection disconnects live nodes first and reports how many were skipped. A new retention policy automatically cleans up closed sessions, and idle SSH connections are now retired automatically.

Remove several stale SSH sessions at once without disconnecting them one by one — select a range in the session manager sidebar, then delete.

📍 In the session manager sidebar, click the first session, then Shift-click the last session in the range to select all sessions between them, then choose Remove to disconnect live nodes and delete them all.

— Clear UI steps and ordering given, but no config key for retention policy. v2.4.1

02 **Prompt cache hit-rate visibility in AI context** Not checked yet

NEW

70

The AI context popover now includes a Prompt Cache section showing cache hit rate (cache reads / input tokens) for the last request and for the session total, plus cache write counts.

Check how well your AI provider's prompt cache is performing to understand token costs across a session.

📍 Open the context popover in the AI chat view and select the Prompt Cache section to see the cache hit rate (cache reads / input tokens) for the last request and the session total, plus cache write volume.

— Exact UI location and formula named, no API surface. v2.4.1

THINNER COVERAGE BELOW

03 **Faithful replay of multi-turn AI tool calls** Not checked yet

IMPROVED

45

Multi-turn AI tool calls now replay exactly as sent to the provider, preserving tool turns across OpenAI-compatible, Anthropic, and Gemini requests.

— Describes behavior and providers, no reproduction steps. v2.4.1

04 **Explicit empty password support for RDP and VNC** Not checked yet

NEW

45

RDP and VNC connections now support saving and syncing explicit empty passwords, rather than treating blank fields as unset.

— Names protocols and behavior, no exact UI field referenced. v2.4.1

05 **Per-connection visual bells and tab highlighting** Not checked yet

NEW

40

Visual bells can now be configured per connection, with background tab highlighting also available.

— Thin description, no config key or toggle path named. v2.4.1

06 **Faster SSH reconnects across split panes** Not checked yet IMPROVED

40

SSH reconnects in split panes across multiple servers no longer wait on the global multi-hop lock, removing a contention point when several panes reconnect simultaneously.

— Explains mechanism removed, no benchmark or config detail. v2.4.1

07 **OLED bloom and CRT terminal effects** Not checked yet NEW

40

The terminal now shows optional OLED bloom and CRT effects, available across GPU backends.

— Names effects and backend scope, no toggle location given. v2.4.1

08 **Zen Mode optional tab bar** Not checked yet IMPROVED

25

Zen Mode now supports keeping the tab bar visible, rather than hiding it entirely.

— One-line description, no toggle or setting named. v2.4.1

WAS THIS USEFUL?  

NeuML txtai

Feedback ⓘ v9.15.0 NOTES ↗ RANK

Txtai is an embeddings database for semantic search, LLM orchestration, and language model workflows, with sparse, dense, and graph indexes in one store.

> txtai v9.15.0 adds new vector backend options (Milvus Lite and a zvec datatype setting), OpenAI-compatible client providers for both LLM and Vectors, and a flag to preserve chain-of-thought tokens in LLM/RAG responses.

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

01 **stripthink flag in LLM and RAG API endpoints** Not checked yet

NEW 70

Pass `stripthink=False` through the LLM and RAG API endpoints to preserve chain-of-thought reasoning tokens in responses.

— Names exact flag and endpoints, directly usable v9.15.0

THINNER COVERAGE BELOW

02 **Milvus Lite and zvec datatype backend options** Not checked yet

NEW 50

txtai now supports Milvus Lite as a sparse ANN backend for vector indexing, and adds a `datatype` setting to the `zvec` backends to control vector storage precision.

— Names both settings but no usage example or defaults given v9.15.0

03 **OpenAI client providers for LLM and Vectors** Not checked yet NEW

45

txtai adds OpenAI client providers for both LLM and Vectors, enabling use of OpenAI-compatible APIs for language model inference and embedding generation.

— Explains purpose but no config keys or setup steps given v9.15.0

WAS THIS USEFUL? 👍 👎

◆ Document Parsing

Docling

Feedback ⓘ v2.152.0 NOTES ↗ RANK

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

> Docling's latest release adds native MLX acceleration for Nemotron 3.5 Parse on Apple Silicon, structures audio/video transcripts and slide media as document tracks, improves multilingual heading detection, and lets the server auto-select a parsing pipeline.

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

01 **MLX support for Nemotron 3.5 Parse** Not checked yet NEW

52

The VLM pipeline now has native MLX support for Nemotron 3.5 Parse, enabling accelerated on-device document parsing on Apple Silicon.

– Names model and platform but no flag or command shown v2.152.0

02 **Automatic default pipeline selection in server** Not checked yet

IMPROVED

40

The server can now choose the default pipeline automatically, removing the need for explicit pipeline configuration in service deployments.

– Behavior change described but no config key or flag named v2.152.0

03 **Audio/video transcript tracks in document model** Not checked yet

NEW

38

Docling now converts audio/video transcripts and slide media into tracks within the document representation, structuring timed media content alongside other document elements.

– Describes the capability but no schema or API detail given v2.152.0

04 **Multilingual heading-hierarchy keyword detection** Not checked yet

IMPROVED

33

Heading-hierarchy detection now reads keyword markers from non-English languages, broadening accurate structure extraction across multilingual documents.

– Explains the change but no language list or mechanism v2.152.0

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

Comet Opik

Feedback ⓘ 2.3.0 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik shipped a generative UI engine for LLM application frontends, a cursor-based stream for dataset experiment items, default-on sandboxing for its Python executor, and a small Experiments page filtering discoverability tweak.

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

01 **Generative UI engine for LLM frontends** Not checked yet NEW

75

Opik now publishes a generative UI engine package to npm and PyPI for use in LLM application frontends. The underlying generative UI foundation includes a views API, an iframe host, an HTML contract, and authoring packages.

– Names npm/PyPI packages and API pieces but no usage snippet 2.3.0 2.2.154

THINNER COVERAGE BELOW

02 **Cursor-based stream for dataset experiment items** Not checked yet

NEW

45

Opik adds a cursor stream for a dataset's experiment items on the backend, enabling paginated or real-time iteration over large experiment result sets.

– Describes mechanism but no endpoint or parameter named 2.3.0

03 **Default process isolation for Python executor** Not checked yet

IMPROVED

40

Opik now turns on Python backend executor hardening and process isolation by default, strengthening the sandboxing of evaluated code.

– Behavior change stated, no config flag or limit given 2.3.0

04 **Add filter chip on Experiments page** Not checked yet IMPROVED

40

Opik now shows an Add filter chip on the Experiments page when no filter is active, making it easier to discover filtering.

– Clear UI location but no further mechanism detail 2.2.154

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ arize-phoenix-client-v3.8.0 NOTES ↗

RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix added programmatic secrets management to its Python client and improved acceptance criteria reporting with eligible-run coverage metrics.

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

01 **Eligible run count in acceptance criteria results** Not checked yet

IMPROVED 65

Acceptance criteria results now expose `eligibleRunCount` (non-skipped runs in the suite) alongside `sampleCount` (runs that logged the criterion's annotation), and the reporter displays coverage as '70 of 100 runs'.

— Names exact fields and example output format `@arizeai/phoenix-client@8.0.3`

THINNER COVERAGE BELOW

02 **Secrets resource in Python client** Not checked yet NEW

35

A new secrets resource was added to the Python client, allowing secrets to be managed programmatically.

— Names surface but no mechanism or usage detail `arize-phoenix-client-v3.8.0`

WAS THIS USEFUL? 👍 👎

UK AI Security Institute Inspect AI

Feedback

0.3.292 CODE ↗

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

Inspect AI adds `--delete-shards`, per-task `max_samples`, `SandboxTimeoutReason` classification, an expanded recover_eval_log(), and a view_image() tool for sandboxes.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 0.3.292
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.292
```

Delete merged log shards after a merge completes using the `--delete-shards` flag on the merge command. Not checked yet

Classify sandbox timeouts by cause using the `reason` field on `SandboxTimeoutError`, typed as `SandboxTimeoutReason = Literal['command', 'no_response', 'unknown']`, exported from `inspect_ai.util`; scorers can now distinguish a command deadline from an unresponsive sandbox. Not checked yet

Let a Task set its own `max_samples` directly, independent of the run-level setting. Not checked yet

Recover crashed eval logs without a sample buffer using recover_eval_log(), now accepting a chosen status, completion time, and log update. Not checked yet

Give models access to image files inside a sandbox with the new view_image() tool. Not checked yet

+1 more

NEW SUPPORTED PLATFORM Gemini 4 Neon

└──▷ BREAKING ON UPGRADE

! exclude_samples() and zero_samples() now return None, and `DispositionResult` is no longer exported; replace any use of `result.applied` or `result.unchanged` with reads from the log.

! `inspect log exclude` and `inspect log zero` no longer accept `--json` and now print a summary instead of per-sample lines; update scripts to read the log or the printed summary.

! Sandbox providers' `task_init`, `task_init_environment`, and `task_cleanup` now run in the launch directory instead of the task directory; a non-Docker provider whose `task_init_environment` returns variables now fails setup with a `PrerequisiteError` — move what `task_init` needs into the sandbox config and pass variables via subprocess(env=...).

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.84.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse's latest release focuses on organization-wide controls for the AI Gateway—spend budgets and multi-region Vertex AI support—alongside organization-scoped model definitions, annotation queue tooling, and several smaller API and UI refinements.

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

01 AI Gateway spend budgets and Vertex AI multi-region support

Not checked yet NEW 62

The AI Gateway now supports organization-wide spend budgets with warning alert thresholds and configurable reset periods, plus support for Vertex AI US and EU multi-region endpoints.

— Names budget mechanics and regions but no exact config path v4.84.0

02 Annotation queue bulk ingestion and score visibility

Not checked yet IMPROVED 62

Adds a `batchCreateAnnotationQueueItems` tool to the MCP server for bulk annotation queue ingestion, and annotation queue items now show all scores rather than a subset.

— Names the MCP tool, a concrete action readers can call v4.84.0

THINNER COVERAGE BELOW

03 Native OTEL direct processing Not checked yet NEW 52

Exposes owned OpenTelemetry batch lifecycle via an opt-in native OTEL direct processing path.

— Describes the mechanism but no exact flag or setting named v4.84.0

04 OpenAI data residency base URL picker Not checked yet IMPROVED 50

LLM connection settings UI now lets users pick OpenAI data residency base URLs from a list instead of entering them manually.

— Names the UI location, a clear navigation starting point v4.84.0

05 Organization-scoped model definitions Not checked yet NEW 45

Model definitions can now be scoped to an organization so models are managed once and apply across all projects, rather than being configured per project.

— Explains benefit but no UI path or config key given v4.84.0

06 **Org-level billable units API endpoint** Not checked yet NEW

43

Adds an unstable org-level API endpoint that returns billable units per project.

— Mentions an API endpoint but omits its path and method v4.84.0

07 **Keyed state for decision evaluators** Not checked yet NEW

35

OpenAI and Gemini decision evaluators now support keyed state.

— States the capability without explaining the mechanism v4.84.0

08 **Intent topic shares in tracing pulse chart** Not checked yet NEW

25

The tracing pulse chart header now shows Intent topic shares.

— Minimal description of a UI addition v4.84.0

09 **Filter value autocomplete in search bar** Not checked yet IMPROVED

20

Filter values are now suggested as you type in the filter search bar.

— Bare description with no scope or location detail v4.84.0

WAS THIS USEFUL?  

Braintrust

Feedback ⓘ updated · Nov 24, 2026 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust's November 2026 release makes Patterns generally available for surfacing recurring agent issues, adds a Decisions API for LLM judges, scheduled digests, and a grouped thread view, and ships a wide set of Terraform provider and SDK changes including opt-in audio capture on OpenAI spans, trace continuation via `bt trace run --parent`, and a LiveKit Agents span rename.

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

01 **Audio attachment opt-in for OpenAI spans** Not checked yet BREAKING

95

OpenAI spans no longer attach audio by default; set `BRAINTRUST_CAPTURE_USER_AUDIO_ATTACHMENTS=true` (input audio) or `BRAINTRUST_CAPTURE_AGENT_AUDIO_ATTACHMENTS=true` (generated audio) in the Python SDK v0.52.0 to restore audio attachments on transcription, translation, speech, and chat spans. Import `AudioRecordingOptions` and `AudioEncoding` from `braintrust.audio` to control voice recording settings such as segment length, file format, and Opus bitrate.

— Names exact env vars, SDK version, and import path snapshot-20261124

02 **Continue traces with `--parent` in `bt trace run`** Not checked yet NEW

80

Continue a trace from an existing run by passing `--parent` with the value returned by an SDK span's export() to `bt trace run`, attaching a coding-agent session below that span instead of starting a new trace.

— Exact flag and command given, mechanism explained snapshot-20261124

03 **Versioned secrets and env vars in Terraform provider**

Not checked yet BREAKING

78

Add a `secret_version` field alongside `secret` in `braintrust_ai_secret`, and a `value_version` field alongside `value` in `braintrust_env_var`; increment the version field to write a new value. `braintrust_ai_secret` now requires `secret_version` and `braintrust_env_var` requires `value_version` — changing only the value no longer updates it, and references to `secret` or `value` return `null`.

— Exact fields and before/after behavior named, no migration command snapshot-20261124

04 **LiveKit Agents span renaming and TTS timing move** Not checked yet

78

BREAKING

LiveKit Agents now uses `livekit.llm_response` for LLM calls it traces directly (including OpenAI Responses and realtime calls), replacing `LLM request` and `Realtime request`; update saved queries, filters, and scorers accordingly. For text-to-speech calls traced directly by LiveKit, `time_to_first_audio` moved from the `tts` span to `time_to_first_token` on the `livekit.tts` child span.

— Names exact span types and fields but no automated migration tool snapshot-20261124

---

05 **Patterns GA for surfacing recurring agent issues** Not checked yet

NEW

75

Patterns is now generally available, investigating agent traces and application logs to surface recurring problems, user frustration, incomplete workflows, unnecessary model calls, and cost or latency shifts. Pattern rules let you narrow Patterns runs to specific traces with a SQL filter and list problems for Loop to watch for. Saved Patterns views can also be managed as Terraform code by setting `view_type` to `patterns` in `braintrust_view`.

Manage a saved Patterns view as Terraform code so it is version-controlled and reproducible across environments.

hcl

```
resource "braintrust_view" "patterns_overview" {
  name      = "Patterns Overview"
  view_type = "patterns"
  project_id = var.project_id
}
```

— Covers GA scope, pattern rules mechanism, and Terraform management snapshot-20261124

---

06 **Terraform provider version requirement and org_automation removal**

Not checked yet

75

BREAKING

Terraform provider v1.0.0 requires Terraform or OpenTofu 1.11.1 or later. The `braintrust_org_automation` resource is removed because the Braintrust API no longer supports it; delete its blocks from your configuration, then run `terraform state rm` on each address before running `terraform plan`.

— Gives exact version floor and removal migration steps snapshot-20261124

---

07 **Metadata argument for Terraform AI secrets and env vars**

Not checked yet NEW 65

Add a `metadata` argument to `braintrust_ai_secret` and `braintrust_env_var` in Terraform provider v1.1.0 to configure AI provider settings as a JSON object loadable from a JSON or YAML file.

— Names resources and config format but no example given snapshot-20261124

08 **Simplified `bt mcp run` invocation** Not checked yet IMPROVED 60

Call a Braintrust MCP server tool directly from the CLI without supplying a telemetry argument, reducing boilerplate in scripted or automated workflows, e.g. `bt mcp run my-mcp-server my_tool --arg1 value1`.

Call a Braintrust MCP server tool directly from the CLI without supplying a telemetry argument, reducing boilerplate in scripted or automated workflows.

```
$ bt mcp run my-mcp-server my_tool --arg1 value1
```

— Runnable command shown but little mechanism explained snapshot-20261124

THINNER COVERAGE BELOW

09 **Ephemeral service token resource in Terraform** Not checked yet NEW 55

Add an ephemeral `braintrust_service_token` resource in Terraform to create a service token without storing its key in state or plan files.

— Names the resource and benefit but no usage example snapshot-20261124

10 **OpenCode 2 tracing in bt CLI** Not checked yet NEW 23

The `bt` CLI adds tracing support for OpenCode 2.

— Named in summary only, no command or config shown snapshot-20261124

11 **Decisions API for LLM judges** Not checked yet NEW 21

Braintrust adds a Decisions API for working with LLM judges.

— Only named in summary, no endpoint or mechanism given snapshot-20261124

12 **Scheduled digests** Not checked yet NEW 10

Braintrust adds scheduled digests.

— Mentioned only in summary with no further detail snapshot-20261124

13 **Grouped thread view** Not checked yet NEW 10

Braintrust adds a grouped thread view.

— Mentioned only in summary with no further detail snapshot-20261124

└──▷ **BREAKING ON UPGRADE**

! LiveKit Agents now uses `livekit.llm_response` for LLM calls it traces directly (including OpenAI Responses and realtime calls), replacing `LLM request` and `Realtime request`; update saved queries, filters, and scorers accordingly.

! For text-to-speech calls traced directly by LiveKit, `time_to_first_audio` moved from the `tts` span to `time_to_first_token` on the `livekit.tts` child span.

! OpenAI spans no longer attach audio by default; set `BRAINTRUST_CAPTURE_USER_AUDIO_ATTACHMENTS=true` or `BRAINTRUST_CAPTURE_AGENT_AUDIO_ATTACHMENTS=true` to restore audio attachments on transcription, translation, speech, and chat spans.

! Terraform provider v1.0.0 requires Terraform or OpenTofu 1.11.1 or later.

! `braintrust_ai_secret` now requires `secret_version` alongside `secret`, and `braintrust_env_var` requires `value_version` alongside `value`; changing only the value no longer updates it, and references to `secret` or `value` return `null`.

! The `braintrust_org_automation` Terraform resource is removed because the Braintrust API no longer supports it; delete its blocks from your configuration, then run `terraform state rm` on each address before running `terraform plan`.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ **AI Security Posture & Discovery**

Harmonic Security

Feedback ⓘ

Harmonic Security Announces New Integration with CrowdStrike Falcon® Next-Gen SIEM

NOTES ↗

Harmonic Security detects shadow AI and prevents sensitive-data leaks across employee web, desktop, and agent workflows.

NEW INTEGRATION CrowdStrike Falcon Next-Gen SIEM integration

WAS THIS USEFUL? 👍 👎

◆ AI Runtime Defense & Guardrails

FailproofAI

Feedback ⓘ v1.0.16 NOTES ↗ RANK

FailproofAI provides observability and enforcement for AI agent harnesses, capturing every run and blocking dangerous tool calls before they execute.

> FailproofAI's v1.0.16 release adds local daemon logging, automated hook-gap detection, opt-out-controlled log uploads to Cloud, machine-fact reporting, and tighter linking of policy checks to the originating tool call across six agent harnesses.

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

01 **Daemon log upload to Cloud with opt-out toggle** Not checked yet

NEW 85

The daemon now uploads its own log to Cloud in two modes: a roll-up of distinct warnings and errors at most every 5 minutes, and a 5-minute window around an error at most every 15 minutes; it redacts secrets and removes the home folder and machine token before upload. Org admins can disable this via the `FAILPROOFAI_NO_LOG_UPLOAD` environment variable.

After upgrading, verify the daemon is at 1.0.16 and check how many events are queued for Cloud delivery.

```
$ failproofai config --status
```

— Names both upload modes, redaction behaviour, and the exact opt-out env var. v1.0.16

02 **Local daemon log file** Not checked yet NEW 70

A new daemon log is written at `~/.failproofai/logs/daemon.log` on every OS, owner-only, rotated at 10 MB with 3 files kept, with a timestamp and level on every line.

— Exact path and rotation parameters given, but no command shown for it. v1.0.16

03 **Hook-gap detection for missing tool-call data** Not checked yet

NEW 60

The daemon checks every 5 minutes whether hook data is still flowing and logs an ERROR (surfaced in Cloud) when an agent's tool calls are not reaching failproofai's hooks, when hook batches are stuck for 15 minutes, or when Cloud is refusing hook batches.

— Clear mechanism and thresholds, but no reader action to take. v1.0.16

04 **Checks linked to originating tool call across agent harnesses**

Not checked yet IMPROVED 60

Checks in Cloud now sit under the tool call they governed, carrying the agent's own tool-call id from Claude, Devin, Factory, Pi, Hermes, and OpenClaw.

— Names six supported harnesses but no concrete UI or API detail. v1.0.16

THINNER COVERAGE BELOW

05 **Machine facts reported on policy poll** Not checked yet NEW

45

The daemon now reports machine facts — daemon version, CLI version, OS and architecture — to Cloud on every policy poll.

— States what is reported but no destination or use detailed. v1.0.16

WAS THIS USEFUL? 👍 👎

◆ AI Red Teaming & Testing

NVIDIA SkillSpector

Feedback ⓘ v2.13.7 NOTES ↗

SkillSpector scans AI agent skills for vulnerabilities and malicious patterns before installation.

SkillSpector adds `--yara-rules-dir` support during baseline generation and an `include_report=false` MCP option, plus broader `pyproject.toml` dependency parsing.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.13.7
https://github.com/NVIDIA/SkillSpector.git
# already have the repo? check out this version:
$ git checkout v2.13.7
```

Support `--yara-rules-dir` during baseline generation so the same additional YARA rules apply to both scans and the baselines they compare against. Not checked yet

MCP callers can now pass `include_report=false` to omit the rendered report from the tool response. Not checked yet

Read dependency declarations from Poetry, PDM, Hatch, and uv tables in `pyproject.toml` to broaden dependency coverage. Not checked yet

Reports now include the sanitised acquisition origin and list built-in recursive root exclusions such as `.git/` and `.venv/` without counting those directories against coverage. Not checked yet

Point RP3 findings to the manifest's version field and include a bounded version snippet for clearer remediation context. Not checked yet

UPDATED DETECTION MCP tool-poisoning findings tagged with MITRE ATLAS AML.T0110

NEW SUPPORTED PLATFORM MiniMax regional provider support

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ updated · Nov 24, 2026 NOTES ↗ RANK

The Claude Developer Platform is the API for building applications on Anthropic's Claude models that generate, analyze, and automate work.

> Anthropic introduced an Enterprise Frontier Safeguards beta offering Zero Data Retention via encrypted Sealed Safety Data, alongside new organization webhook events for monitoring external storage health.

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

01 **External storage webhook events for organizations** Not checked yet

NEW 60

Organization webhooks now fire `external_storage.failed` when customer-owned bucket storage fails a daily check and `external_storage.recovered` when it passes again.

– Names exact webhook event types a reader can subscribe to snapshot-20261124

THINNER COVERAGE BELOW

02 **Enterprise Frontier Safeguards beta with Zero Data Retention**

Not checked yet NEW 57

New Enterprise Frontier Safeguards (EFS) beta on Claude Platform, Claude Platform on AWS, and Claude Models on Google Cloud (Anthropic managed) gives eligible organizations Zero Data Retention, with prompts and responses stored only as encrypted Sealed Safety Data in the customer's own cloud account rather than on Anthropic-managed storage.

– Explains mechanism and platforms but no setup steps given snapshot-20261124

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

OpenJev-1.1

Feedback ⓘ released · Nov 24, 2026 MODEL ↗

OpenJev-1.1 from OpenJev was released on 23 Nov 2026. Open weights on Hugging Face. Not checked yet

WAS THIS USEFUL? 👍 👎

NVIDIA Nemotron 3.5 Content Safety

Feedback ⓘ

retirement scheduled · Nov 23, 2026 MODEL ↗

Nemotron 3.5 Content Safety is a multimodal model from NVIDIA.

OpenRouter lists a retirement date of 30 Nov 2026 for nvidia/nemotron-3.5-content-safety, first seen 23 Nov 2026. Not checked yet

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · November 24, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue