THE AI TOOLCHAIN

NO. 030

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 18, 2026 · EVERY DAY

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

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 292 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 50 tools

▾ // THE BRIEF HIDE

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

Three unrelated things stand out. Braintrust's Patterns runs scheduled investigations over your traces and surfaces recurring failure modes ut anyone opening a trace viewer. Weaviate 1.39's 4-bit Rotational

vim ? help

Quantization cuts vector memory while keeping recall. And Modal now caps spend per environment, so a runaway job hits a ceiling instead of an invoice.

OPERATE

Find the recurring failure in your traces without reading traces

Braintrust's Patterns is a scheduled Loop-powered job that investigates traces for repeating problems, replacing the weekly ritual of hand-sampling logs to find what keeps breaking. LangSmith's annotation queues now take conversation threads as well as individual runs, so a multi-turn failure gets reviewed as the conversation it actually was rather than as disconnected calls.

Braintrust · LangSmith

DEPLOY

Serve the same index on less RAM, and pick a GPU shape you know will fit

Weaviate 1.39's 4-bit Rotational Quantization with optional centering shrinks the memory footprint while improving recall, plus SIMD encoding and CPU prefetching that speed search and imports — the usual quantization tradeoff of paying recall for RAM moves in your favor. Fireworks validates deployment shapes across firectl, the Create Deployment API and the Python SDK, with a flag to auto-select and a subcommand to list compatible shapes, so an incompatible GPU config fails at request time instead of after a failed rollout.

Weaviate · Fireworks AI

OPERATE

Put a hard ceiling on what an environment can spend

Environment-level spend budgets mean a stuck agent loop or a misconfigured batch stops at a number you set rather than being discovered on the bill. The same release GAs VM Sandboxes and adds Sandbox Sidecars for isolated compute, and cuts Function call latency on a faster I/O path.

Modal

GOVERN

Stop an agent's outbound call at a policy boundary instead of auditing it later

> Agent Safe Pipeline's HTTP proxy gateway enforces ALLOW/BLOCK/ESCALATE decisions from the Decionis control plane, with TLS egress sealing, cryptographic evidence chaining and a boundary self-test command — the approval step becomes an endpoint rather than a code review. Gram adds Pi agent install, session capture and policy enforcement plus an expanded Shadow AI inventory, which is how you find the agents nobody registered.
>
> agent-safe-pipeline

BUILD

◆ AI Coding Agents

Warp

Feedback ⓘ 2026.09.16 (v0.2026.09.16.08.27) NOTES ↗ RANK

Warp is an AI terminal that runs commands and helps developers build software.

> Warp's latest release adds a CLI subcommand for creating container registry credential secrets, terminal-level triggering of fzf and atuin, and a native shell-completions fallback, alongside new workspace/team membership controls, Windows window-backdrop styling, and smaller improvements to build-tool discovery and command history search.

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

01 CLI subcommand for registry credential secrets NEW

75

Adds `oz secret create docker-registry` to create container registry credential secrets directly from the terminal, for use in orchestrated agent runs such as pulling from a private Docker registry.

Create a container registry credential secret for use in orchestrated agent runs, such as pulling from a private Docker registry.

```
$ oz secret create docker-registry
```

Create a container registry credential secret for use in orchestrated agent runs without leaving the terminal.

```
$ oz secret create docker-registry
```

— Exact runnable command with clear use case named. 2026.09.16 (v0.2026.09.16.08.27)

02 fzf and atuin triggers inside Warp NEW

65

Supports triggering `fzf` (via `ctrl-r` or `ctrl-t`) and `atuin` directly from within Warp when configured with your shell.

— Names exact keybindings but not the shell config steps.

2026.09.16 (v0.2026.09.16.08.27)

03 Native shell completions fallback NEW

60

Adds native shell completions alongside Warp's built-in completion specs, falling back to native completions when Warp's built-in specs have no suggestions; configurable in

`Settings > Features > Terminal Input`.

Enable native shell completions to fill gaps when Warp's built-in specs return no suggestions — or override the fallback behavior entirely.

> 📍 Open Settings › Features › Terminal Input and adjust the native shell completions setting.

Enable native shell completions as the primary suggestion source rather than the fallback, via Warp settings.

> 📍 In Warp, go to Settings › Features › Terminal Input and adjust the native shell completions behavior.

— Explains fallback mechanism and exact settings path. `2026.09.16 (v0.2026.09.16.08.27)`

THINNER COVERAGE BELOW

04 **Workspace and team membership controls** NEW — 55

Workspace admins can now remove members from a workspace directly from team settings; native workspace members can leave a team without losing workspace access; and native workspace users can browse and join another open team from Teams settings.

— Three related actions named with UI paths, no endpoints.

`2026.09.16 (v0.2026.09.16.08.27)`

05 **Windows backdrop materials** NEW — 45

[Windows] Adds Materials support for window backdrops, configurable in `Settings > Appearance > Window > Backdrop`.

— Names exact settings path, no further mechanism given. `2026.09.16 (v0.2026.09.16.08.27)`

06 **Build tool discovery in nested repositories** IMPROVED — 30

Discovers and caches build tools in nested repository directories.

— Bare description with no mechanism or scope given. `2026.09.16 (v0.2026.09.16.08.27)`

07 **Command history search ranking improvements** IMPROVED — 30

Improves command history search ranking using match quality and usage signals.

— Names ranking signals but no concrete detail on use. `2026.09.16 (v0.2026.09.16.08.27)`

08 **Tab group persistence for legacy launch configs** IMPROVED — 20

Adds tab group persistence to legacy launch configs.

— Single unelaborated sentence with no mechanism or path. product docs

└──▷ ALSO FROM THESE RELEASES

2026.09.16 (v0.2026.09.16.08.27) NOTES ↗ LATEST STABLE

Warp adds fzf/atuin trigger support, native shell completions, and Windows backdrop materials in this release.

└──▷ TRY IT

Create a container registry credential secret for use in orchestrated agent runs, such as pulling from a private Docker registry.

```
$ oz secret create docker-registry
```

Enable native shell completions to fill gaps when Warp's built-in specs return no suggestions — or override the fallback behavior entirely.

📍 Open Settings › Features › Terminal Input and adjust the native shell completions setting.

Adds `oz secret create docker-registry` CLI subcommand to create container registry credential secrets.

Supports triggering `fzf` (via `ctrl-r` or `ctrl-t`) and `atuin` directly from within Warp when configured with your shell.

Adds native shell completions alongside Warp's built-in completion specs, falling back to native completions when Warp's built-in specs have no suggestions; configurable in Settings > Features > Terminal Input.

[Windows] Adds Materials support for window backdrops, configurable in Settings > Appearance > Window > Backdrop.

Workspace admins can now remove members from a workspace directly from team settings.

+4 more

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.156.0-alpha.2 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 experimental voice conversations in the TUI alongside new daemon update controls, Windows MXC sandbox permission management, and MCP authentication improvements, while removing the `thread/rollback` API.

└──▷ **WHAT SHIPPED · 17 FEATURES** most completely described first ? what's the number?

01 **Voice conversations in TUI** NEW 84

The TUI gained experimental voice conversations, opened via `/experimental`, and the `/voice settings` slash command to choose a voice for future conversations; voice conversations are now enabled by default. The experience includes live transcripts, microphone controls, a mute shortcut, and a split-flap animation on supported builds, backed by bundled Linux and Windows native voice runtimes for improved audio reliability.

Switch the voice used for all future TUI voice conversations without leaving the session.

```
$ /voice settings
```

– Names the enabling command and mechanism but no limits rust-v0.155.0 rust-v0.155.0-alpha.9.2

02 **Windows sandbox permissions and MXC controls** NEW 80

The Windows MXC sandbox gained managed network policy support, and the Windows offline sandbox now blocks non-loopback inbound traffic. MXC selection is configured via `config.permissions.windows_sandbox_type` set to `SandboxType::WindowsMxc`, and filesystem reads and writes are now routed through their own sandbox permissions, separating read and write access controls.

– Names exact config key and sandbox type value rust-v0.155.0-alpha.9.2 rust-v0.155.0-alpha.17

03 **Daemon update management** NEW 76

Codex CLI added an explicit `codex app-server daemon update` command for on-demand daemon updates, with configurable automatic update schedules, a `/daemon`

menu in the TUI for managing local background server updates, and daemon auto-resume after restarts.

Trigger a daemon update on demand without waiting for the automatic schedule.

```
$ codex app-server daemon update
```

— Gives a runnable command plus menu, light on schedule detail `rust-v0.155.0` `rust-v0.156.0-alpha.2`

04 **MCP authentication and login improvements** NEW 61

Codex CLI added trusted enterprise MCP auth configuration, manual callback input to MCP OAuth login flows, and Touch ID (Secure Enclave) verification for MCP requests in local TUI sessions on supported Macs.

— Names three auth surfaces but no config keys given `rust-v0.155.0` `rust-v0.155.0-alpha.9.2`

05 **AWS credential export for Amazon Bedrock** NEW 60

Codex CLI added AWS credential export commands for Amazon Bedrock, with caching, expiration-based refresh, and authentication recovery.

— Describes mechanism but no exact command name given `rust-v0.155.0`

THINNER COVERAGE BELOW

06 **Multi-agent V2 tool catalog descriptions** NEW 57

Catalog descriptions are now supported for all multi-agent V2 tools, allowing model catalogs to override `spawn_agent` descriptions.

— Names the overridable field but no example given `rust-v0.155.0-alpha.17`

07 **Agents overview and worktree management** IMPROVED 55

Worktrees are now enabled by default for local daemon sessions. The agents overview gained task hiding, archiving, and deletion, along with worktree ownership details and confirmed deletion of clean managed worktrees.

— Lists UI actions without exact navigation paths `rust-v0.155.0` `rust-v0.155.0-alpha.9.2`

08 **Guardian action review improvements** IMPROVED 50

Guardian's action review limit was raised to 200,000 bytes, and delegation reviews now include sender user messages for more complete review context.

— Gives a concrete limit but no config path `rust-v0.155.0` `rust-v0.155.0-alpha.17`

09 **TUI rendering and display improvements** IMPROVED 50

The TUI now renders Markdown in agent overview task details, applies consistent theme-based thread colors, honors system reduced-motion preferences, expands Unicode math rendering with accents, symbols, and delimiters, and renders Mermaid diagrams.

– Lists rendering additions without usage steps `rust-v0.155.0-alpha.9.2` `rust-v0.155.0-alpha.17` `rust-v0.156.0-alpha.2`

10 **User verification for Codex Desktop sessions** NEW 45

Local Codex Desktop sessions now enable user verification when `experimentalApi` is enabled.

– Names the enabling flag but no further detail `rust-v0.155.0-alpha.9.2`

11 **App-server API status exposure** IMPROVED 42

The app-server API now exposes advertised MCP server capabilities in status responses and exposes disabled plugin settings.

– Brief description, no field names or endpoint paths `rust-v0.155.0-alpha.9.2`

12 **Live streaming reasoning summaries in TUI** NEW 40

The TUI status row now shows live streaming reasoning summaries and completion timestamps after successful turns.

– Describes behavior without a config or command `rust-v0.155.0`

13 **Opt-in overhead timing for code-mode responses** NEW 37

Codex CLI added opt-in overhead timing to code-mode responses for measuring request latency.

– Describes purpose but not how to enable it `rust-v0.155.0-alpha.17`

14 **Removal of thread/rollback API** BREAKING 35

The `thread/rollback` API has been removed.

– Names removed endpoint but no migration guidance `rust-v0.155.0-alpha.9.2`

15 **Hidden HTTP/3 TCP tunnel command** NEW 23

Codex CLI added a hidden HTTP/3 TCP tunnel command.

– Command exists but name and usage not disclosed `rust-v0.156.0-alpha.2`

16 **Experimental rollout compression endpoint** NEW 21

Codex CLI added an experimental rollout compression endpoint.

– No endpoint path or usage details given `rust-v0.156.0-alpha.2`

17 Warning for ignored configuration settings IMPROVED

18

Codex CLI now warns about ignored configuration settings.

– No specifics on which settings or where warning appears `rust-v0.155.0-alpha.9.2`

└──▷ **BREAKING ON UPGRADE**

! The `thread/rollback` API has been removed.

`rust-v0.156.0-alpha.2` NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI gains Mermaid diagram rendering in the TUI, daemon auto-resume after restarts, and a `/daemon` update menu.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch rust-v0.156.0-alpha.2
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.156.0-alpha.2
```

Adds a `/daemon` menu in the TUI for managing local background server updates.

Adds an experimental rollout compression endpoint.

Adds a hidden HTTP/3 TCP tunnel command.

`rust-v0.155.0-alpha.9.2` NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI alpha adds `/voice settings` for voice selection, TUI voice on by default, worktrees enabled by default, and trusted enterprise MCP auth config.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch rust-v0.155.0-alpha.9.2
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.155.0-alpha.9.2
```

└──▷ **TRY IT**

Switch the voice used for all future TUI voice conversations without leaving the session.

```
$ /voice settings
```

Adds `/voice settings` slash command to choose a voice for future conversations in the TUI.

Enables TUI voice conversations by default.

Enables worktrees by default for local daemon sessions.

Adds trusted enterprise MCP auth configuration.

Adds manual callback input to MCP OAuth login flows.

+10 more

└─▷ **BREAKING ON UPGRADE**

! The `thread/rollback` API has been removed.

`rust-v0.155.0` NOTES ↗ CODE ↗ **LATEST** STABLE

Codex CLI gains experimental voice conversations, Touch ID for MCP, configurable daemon updates, and AWS credential commands for Bedrock.

└─▷ **GET THIS VERSION** ▾

```
$ git clone --branch rust-v0.155.0
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.155.0
```

└─▷ **TRY IT**

Trigger a daemon update on demand without waiting for the automatic schedule.

```
$ codex app-server daemon update
```

Adds `codex app-server daemon update` command for explicit daemon update control, with configurable automatic update schedules.

Adds experimental `/voice` conversations via `/experimental` in the TUI, with live transcripts, microphone controls, mute shortcut, and split-flap animation on supported builds.

Adds Touch ID (Secure Enclave) verification for MCP requests in local TUI sessions on supported Macs.

Adds AWS credential export commands for Amazon Bedrock, with caching, expiration-based refresh, and authentication recovery.

Shows live streaming reasoning summaries in the TUI status row and completion timestamps after successful turns.

+2 more

rust-v0.155.0-alpha.17 NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI alpha.17 adds opt-in automatic background server startup, MXC sandbox selection on Windows, and expanded Unicode math rendering in the TUI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rust-v0.155.0-alpha.17
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.155.0-alpha.17
```

Enables MXC selection through Windows sandbox configuration via `config.permissions.windows_sandbox_type` set to `SandboxType::WindowsMxc`.

Adds opt-in automatic background server startup so the app-server can launch without an explicit foreground process.

Adds opt-in overhead timing to code-mode responses for measuring request latency.

Expands Unicode math rendering in the TUI with accents, symbols, and delimiters.

Routes filesystem reads and writes through their own sandbox permissions, separating read and write access controls.

+2 more

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.32.13 NOTES ↗ RANK

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

> Letta Code's biggest change this window is a new backend-switching system letting users move between Letta Cloud and a local backend, alongside a CI code review escalation step and self-contained Python wheels for distribution.

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

01 **Backend selection: cloud vs local** NEW 85

Adds `letta setup` to interactively choose between Letta Cloud and a local backend, saving the choice for future runs. Adds `letta backend cloud` and `letta backend local` subcommands to change the saved default backend, and `--backend cloud` / `--backend local` flags for a one-off override without changing the saved default.

Switch your saved default to local mode when working air-gapped or without a Letta Cloud account.

```
$ letta backend local
```

Permanently switch a workstation from Letta Cloud to a self-hosted local backend.

```
$ letta backend local
```

– Names exact subcommands and flags with runnable examples product docs

THINNER COVERAGE BELOW

02 **Expert escalation step in CI code review** NEW 25

Adds an expert escalation step that runs after code review in CI pipelines.

– No mechanism or trigger conditions given v0.32.13

03 **Self-contained Python wheels for distribution** NEW 25

Letta Code v0.32.13 adds self-contained Python wheels for distribution.

– Only mentioned in summary, no detail on usage v0.32.13

v0.32.13 NOTES ↗ LATEST STABLE

> Letta Code v0.32.13 adds expert escalation in CI code review and self-contained Python wheels for distribution.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.32.13 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.32.13
```

WAS THIS USEFUL?  

GitHub Copilot CLI

Feedback ⓘ v1.0.86 NOTES ↗ RANK

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

> GitHub Copilot CLI's v1.0.86 release lets custom agents opt into repo-level instruction files, improves sandbox policy reporting for local-network access, and fixes session resumption to preserve marketplace plugins and skills.

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

01 Custom agents can read repo instruction files NEW

70

Setting `include-custom-instructions: true` in a custom agent's frontmatter lets it opt into repo instruction files such as `AGENTS.md` or `copilot-instructions.md` for project-specific guidance.

Enable a custom agent to read repo-level instruction files such as `AGENTS.md` or `copilot-instructions.md` for project-specific guidance.

yaml

```yaml
include-custom-instructions: true
```

— Names exact config key and file targets, ready to use v1.0.86

THINNER COVERAGE BELOW

02 Session resume preserves plugins and skills IMPROVED

40

Resuming an active session without plugin-directory, discovery, or working-directory overrides now preserves marketplace plugins and skills after reload.

— Describes before/after behavior but no command shown v1.0.86

03 Sandbox policy reports local-network access setting IMPROVED

35

The `/sandbox` policy now reports local-network access using the configured setting.

— Names the surface but not the mechanism or values v1.0.86

> v1.0.86 NOTES ↗ LATEST STABLE
>
> Custom agents can now opt into repo instruction files via `include-custom-instructions: true` in agent frontmatter.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.0.86 https://github.com/github/copilot-cli.git
# already have the repo? check out this version:
$ git checkout v1.0.86
```

└──▷ **USE IT**

Enable a custom agent to read repo-level instruction files such as `AGENTS.md` or `copilot-instructions.md` for project-specific guidance.

```yaml
include-custom-instructions: true
```

The `/sandbox` policy now reports local-network access using the configured setting.

Resuming an active session without plugin-directory, discovery, or working-directory overrides now preserves marketplace plugins and skills after reload.

WAS THIS USEFUL?  

Anthropic Claude Code

Feedback ⓘ v2.1.275 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.275 adds account-level skills/plugin syncing from claude.ai, a marketplace-aware plugin install subcommand, a send-now hotkey for queued messages, and a batch of VSCode extension UI refinements.

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

01 **`/plugin install --marketplace` subcommand** NEW 85

The `/plugin install <plugin> --marketplace <source>` subcommand installs a plugin from a marketplace source, offering to add the marketplace first if it isn't already registered locally.

Install a plugin from a marketplace that is not yet registered locally — Claude Code will offer to add it first.

```
$ /plugin install <plugin> --marketplace <source>
```

— Exact runnable subcommand with syntax given. v2.1.275

02 **Send-now hotkey for queued messages** NEW 80

A new send-now key (`ctrl+enter` , or `ctrl+x ctrl+s`) interrupts the current turn and sends all queued messages at once; sent and queued messages display in gray until the model receives them.

— Exact keybindings and behaviour described. v2.1.275

03 **claude.ai skills and plugins sync to terminal** NEW 75

Skills and plugins enabled on a signed-in claude.ai account now sync automatically to terminal sessions signed in with that account; the `syncClaudeAiSkills: false` and `syncClaudeAiPlugins: false` config keys allow opting out of this automatic sync.

Prevent claude.ai account skills and plugins from being synced into a terminal session — useful in CI or shared environments where account-level tools should not run.

```yaml
syncClaudeAiSkills: false
syncClaudeAiPlugins: false
```

— Names both config keys and the sync mechanism clearly. v2.1.275

04 **VSCode extension UI updates** IMPROVED 73

The VSCode extension adds viewing, editing and deleting a saved memory inside the Memory dialog; sending an attached image without typing any text; a Retry link in the MCP servers dialog when the server list fails to load; accept and reject buttons under each change in the proposed-change diff tab for reviewing edits change by change; and a `claudeCode.scrollToBottomOnSend` setting to control whether the conversation jumps to the bottom on send.

— Enumerates every named VSCode addition including the setting key. v2.1.275

THINNER COVERAGE BELOW

05 **Startup warning for failed otelHeadersHelper** IMPROVED 55

Claude Code now shows a startup warning when a configured `otelHeadersHelper` fails, surfacing sessions that would otherwise silently export no telemetry.

— Names the config key but no further mechanism. v2.1.275

06 **Signed-in account shown in Claude apps gateway** IMPROVED 55

The signed-in account is now added to the Claude apps gateway sign-in flow: the gateway-named account is confirmed before the credential is saved, and `/status` displays it.

— Names `/status` surface but limited mechanism detail. v2.1.275

07 **New routine button on broken routine links** NEW 40

Claude Code on the web adds a New routine button to the page shown when a routine link no longer resolves.

— Thin, single-sentence description with no further detail. v2.1.275

v2.1.275 NOTES ↗ CODE ↗ LATEST STABLE

Claude Code v2.1.275 adds send-now hotkeys, claude.ai skills/plugins sync with opt-out config keys, and `/plugin install --marketplace`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.1.275 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.275
```

└──▷ **TRY IT**

Install a plugin from a marketplace that is not yet registered locally — Claude Code will offer to add it first.

```
$ /plugin install <plugin> --marketplace <source>
```

Prevent claude.ai account skills and plugins from being synced into a terminal session — useful in CI or shared environments where account-level tools should not run.

```yaml
syncClaudeAiSkills: false
syncClaudeAiPlugins: false
```

Adds `syncClaudeAiSkills: false` and `syncClaudeAiPlugins: false` config keys to opt out of automatic syncing of claude.ai account skills and plugins to terminal sessions.

Adds `/plugin install <plugin> --marketplace <source>` subcommand, which offers to add the marketplace before installing the plugin.

Adds a startup warning when a configured `otelHeadersHelper` fails, so sessions that silently export no telemetry are noticed.

Adds a send-now key (`ctrl+enter`, or `ctrl+x ctrl+s`) that interrupts the current turn and sends all queued messages at once; sent and queued messages show in gray until the model receives them.

Syncs skills and plugins enabled on your claude.ai account to terminal sessions signed in with it.

+7 more

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ desktop-v0.0.31 NOTES ↗ CODE ↗ RANK

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

> Cline desktop v0.0.31 adds SSH-based remote execution so the agent can work directly on remote Linux (and helper-enabled macOS) machines, and switches concurrent sub-agent tasks from sequential to parallel execution.

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

01 **SSH remote environments in desktop app** NEW 93

Cline can connect to and operate on remote hosts over SSH. Host profiles are stored at `~/.cline/data/settings/remote-environments.json` (mode `0600`), recording only the identity-file path, never private-key contents; unknown or changed host keys are rejected. Configuration happens via Settings → Remote, where you add and test a host, then pick it from the environment selector beside the workspace picker — Add project… browses the remote machine and Local disconnects back to local execution. On first connect, a self-contained remote helper is uploaded to `~/.cline/remote/`, requiring no `apt`, `npm`, root access, global CLI install, or public port; the `CLINE_REMOTE_HELPER_BINARY` env var lets you supply a locally built helper, which is required for macOS hosts. Linux x64 and arm64 remote hosts are supported (32-bit Raspberry Pi OS is not). SSH config aliases are honored: leave Port blank to use the alias's own configuration or enter a value to override it, and recent/last-used workspaces are tracked separately per host.

— Names config path, permissions, env var, UI path, platform support desktop-v0.0.31

THINNER COVERAGE BELOW

02 **Concurrent sub-agent execution** IMPROVED 35

Sub-agents spawned within the same step now run concurrently instead of sequentially, so delegating N independent tasks takes as long as the slowest one rather than the sum of all of them.

— Explains mechanism and benefit but no named surface desktop-v0.0.31

desktop-v0.0.31 NOTES ↗ CODE ↗ LATEST STABLE

Cline desktop gains SSH remote execution and concurrent sub-agent parallelism in v0.0.31

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch desktop-v0.0.31
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.31
```

Stores SSH remote-host profiles at `~/.cline/data/settings/remote-environments.json` (mode `0600`), recording identity-file path (never private-key contents); unknown or changed host keys are rejected.

Adds SSH remote-environment support via `Settings → Remote`: add and test a host, then select it from the environment selector beside the workspace picker; `Add project…` browses the remote machine and `Local` disconnects.

Uploads a self-contained remote helper to `~/.cline/remote/` on first connect — requires no `apt`, `npm`, root access, global CLI install, or public port — with a `CLINE_REMOTE_HELPER_BINARY` env var to supply a locally built helper for macOS hosts.

Supports Linux x64 and arm64 remote hosts (32-bit Raspberry Pi OS is not supported; macOS hosts require a locally built helper via `CLINE_REMOTE_HELPER_BINARY`).

SSH config aliases are honored: leave `Port` blank to use the alias's own configuration, or enter a value to override it; recent and last-used workspaces are tracked separately per host.

+1 more

WAS THIS USEFUL? 👍 👎

Block Goose

Feedback ⓘ v1.51.0 NOTES ↗ CODE ↗ RANK

Goose is an open-source AI agent with a desktop app and CLI for coding and other automated tasks.

> Goose v1.51.0 adds an operator allowlist for gateway pairing, two new provider integrations, and an opt-in terminal bell, while removing planning mode and the `create recipe` command from the CLI.

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

01 **Operator allowlist for gateway pairing** NEW 62

Adds `allowed_user_ids` field to gateway platform config, enabling an operator allowlist that restricts which users can pair with a gateway.

– Names exact config field but not full mechanism v1.51.0

THINNER COVERAGE BELOW

02 **CLI removals: planning mode and create recipe** BREAKING 50

Removes planning mode from the CLI (`remove planning mode from the CLI`) and removes the `create recipe` command from the CLI.

– Names exact removed commands but no migration guidance v1.51.0

03 **Opt-in terminal bell in CLI** NEW 45

Adds an opt-in terminal bell that rings when a turn finishes or when approval is needed in the CLI.

– Describes behavior but no flag or setting named v1.51.0

04 **New provider integrations: GPT-live and EUrouter** NEW 35

Adds GPT-live API support as a new provider integration, and adds EUrouter as a declarative provider.

– Names both providers but no setup details given v1.51.0

05 **Toolshim scoping to custom providers** IMPROVED 10

Toolshim scoping is extended to custom providers, per the release summary; no further mechanism or configuration detail is provided.

– Mentioned only in summary, no detail available v1.51.0

└──▷ **BREAKING ON UPGRADE**

! Planning mode has been removed from the CLI (`remove planning mode from the CLI`).

! The `create recipe` command has been removed from the CLI.

v1.51.0 NOTES ↗ CODE ↗ LATEST STABLE

Goose v1.51.0 adds GPT-live API support, EUrouter provider, operator allowlists, opt-in terminal bell, and Toolshim scoping to custom providers.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.51.0 https://github.com/aaif-goose/goose.git
# already have the repo? check out this version:
$ git checkout v1.51.0
```

Adds `allowed_user_ids` field to gateway platform config, enabling an operator allowlist that restricts which users can pair with a gateway.

Adds GPT-live API support as a new provider integration.

Adds EUrouter as a declarative provider.

Adds opt-in terminal bell that rings when a turn finishes or when approval is needed in the CLI.

└──▷ **BREAKING ON UPGRADE**

! Planning mode has been removed from the CLI (`remove planning mode from the CLI`).

! The `create recipe` command has been removed from the CLI.

WAS THIS USEFUL? 👍 👎

sirmalloc ccstatusline

Feedback ⓘ v2.2.30 NOTES ↗

RANK

ccstatusline is a customizable statusline for Claude Code CLI with powerline support, themes, and widget-based configuration.

> ccstatusline's v2.2.30 release focuses on widget refinements: opt-in caching for custom commands, cleaner reset-timer displays, and expanded glyph support for git and jj widgets.

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

01 **TTL caching for custom-command widgets** NEW 65

Adds opt-in TTL caching and timeout enforcement for `custom-command` widgets so repeated invocations can reuse stale output instead of re-running the command on every render.

— Names the widget and mechanism but no config key given v2.2.30

THINNER COVERAGE BELOW

02 **Reset-timer widgets hide empty placeholder** IMPROVED 45

Reset-timer widgets can now hide their no-data placeholder when no usage data is available, reducing visual noise in the status line.

— Describes behavior but no toggle or setting named v2.2.30

03 **Symbol slots for git and jj widgets** IMPROVED 45

The remaining git and jj widgets gain symbol slots for custom glyphs, bringing them in line with other widget types that already support custom glyphs.

— States the change but no example of configuring symbols v2.2.30

v2.2.30 NOTES ↗ LATEST STABLE

v2.2.30 adds opt-in TTL caching for custom commands and lets reset timers hide their no-data placeholders.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.2.30
https://github.com/sirmalloc/ccstatusline.git
# already have the repo? check out this version:
$ git checkout v2.2.30
```

Adds opt-in TTL caching and timeout enforcement for `custom-command` widgets so repeated invocations can reuse stale output instead of re-running the command on every render.

Lets reset-timer widgets hide their no-data placeholder when no usage data is available, reducing visual noise in the status line.

Gives the remaining git and jj widgets symbol slots, bringing them in line with other widget types that already support custom glyphs.

WAS THIS USEFUL?  

Amazon Kiro

Feedback ⓘ

CLI: Fullscreen Chat and Streamlined Sessions: Expand chat to a dedicated terminal scree

BLOG ↗ RANK

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

> Kiro's biggest moves this window are model-side: GPT-5.6 Sol/Terra/Luna and the new Claude Fable 5.1 both gained 1M-token context windows for enterprise and paid tiers, while the CLI picked up a dedicated fullscreen chat mode, a redesigned session dashboard, and reliability work for long-running V3 sessions.

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

01 1M-token context for GPT-5.6 Sol, Terra, Luna IMPROVED

85

GPT-5.6 Sol, Terra, and Luna context windows are extended from 272K to 1M tokens across the IDE, CLI, and Web, enabling full-codebase ingestion in a single request without chunking or summarization. Requests exceeding 272K tokens are billed at double the short-context multiplier (Sol 4.4x→8.8x, Terra 2.2x→4.4x, Luna 1.1x→2.2x) for Pro, Pro+, Pro Max, and Power customers.

Access the expanded 1M-token GPT-5.6 models in Kiro Web after the rollout reaches your account.

> 📍 In the Kiro Web interface, refresh your browser, then open the model selector and choose GPT-5.6 Sol, Terra, or Luna to use the 1M token context window.

— Exact token limits and billing multipliers per model, UI steps given

launch-20260914-08fe5040

02 Fullscreen chat mode in CLI NEW

80

The `/fullscreen` command moves the current conversation to a dedicated terminal screen with its own scrollable view, independent of shell history; typing it again returns inline. It's available in interactive TUI and Lite sessions on both V2 and V3, and a `Start fullscreen` toggle under `/settings` > Display can auto-open future interactive TUI sessions in fullscreen mode.

Switch to fullscreen mid-session to get a scrollable, distraction-free view when working through a long agentic task.

```
$ /fullscreen
```

Enable fullscreen by default for all future interactive TUI sessions so you never have to type `/fullscreen` manually.

> 📍 In the Kiro TUI, type `/settings`, navigate to Display, and turn on `Start fullscreen`.

— Names exact command and settings path with clear mechanism

CLI: Fullscreen Chat and Streamlined Se…

---

03 **KIRO_SKIP_BINARY_PINNING env var** NEW — 70

Setting `KIRO_SKIP_BINARY_PINNING=1` keeps the installed CLI binary at a fixed path, for storage-constrained environments where binary replacement during a session would be a problem.

Run Kiro on a device with a fixed install path (e.g. a read-only volume or embedded environment) where binary replacement during a session would break the process.

```
$ KIRO_SKIP_BINARY_PINNING=1 kiro
```

— Exact env var with runnable command, thin on broader context

CLI: Fullscreen Chat and Streamlined Se…

---

04 **Redesigned /sessions dashboard (V3)** IMPROVED — 65

The V3 `/sessions` dashboard separates search, filter, sort, and grouping controls from the session list, presenting a flat list ordered by last use, with sorting by last use, session name, or message count, and persistent selection memory.

— Describes new UI structure but no exact navigation steps given

CLI: Fullscreen Chat and Streamlined Se…

THINNER COVERAGE BELOW

---

05 **V3 session reliability: large results and compaction indicator**

IMPROVED — 50

In V3, Kiro now saves large tool results with the session and shows the model a short preview, reducing context use without losing the full output. It also keeps the active response and Thinking indicator visible while automatic compaction runs, with queued prompts continuing afterward.

— Explains mechanism but no named flags or config to act on

CLI: Fullscreen Chat and Streamlined Se…

CLI: Fullscreen Chat and Streamlined Sessions: Expand chat to a dedicated terminal scr

BLOG ↗ LATEST STABLE

Kiro CLI 2.22 adds fullscreen chat via `/fullscreen`, a redesigned `/sessions` dashboard, and a `KIRO_SKIP_BINARY_PINNING` env var for fixed-path installs.

└──▷ **TRY IT**

Switch to fullscreen mid-session to get a scrollable, distraction-free view when working through a long agentic task.

```
$ /fullscreen
```

Enable fullscreen by default for all future interactive TUI sessions so you never have to type `/fullscreen` manually.

📍 In the Kiro TUI, type `/settings`, navigate to Display, and turn on `Start fullscreen`.

Run Kiro on a device with a fixed install path (e.g. a read-only volume or embedded environment) where binary replacement during a session would break the process.

```
$ KIRO_SKIP_BINARY_PINNING=1 kiro
```

Adds `KIRO_SKIP_BINARY_PINNING=1` environment variable for storage-constrained environments where the installed binary remains at a fixed path.

Adds `/fullscreen` command to move the current conversation to a dedicated terminal screen with its own scrollable view, independent of shell history; type again to return inline — available in interactive TUI and Lite sessions on both V2 and V3.

Adds a `Start fullscreen` toggle under `/settings` > Display to automatically open future interactive TUI sessions in fullscreen mode.

Redesigns `/sessions` dashboard (V3) to separate search, filter, sort, and grouping controls from the session list, with a flat list ordered by last use, sorting by last use, session name, or message count, and persistent selection memory.

Adds large tool result handling in V3: Kiro saves large results with the session and gives the model a short preview, reducing context use without losing full output.

+1 more

announced · Sep 16, 2026 BLOG ↗

Claude Fable 5.1 with 1M context window and updated cybersecurity safeguards now available to Kiro enterprise customers

└──▷ **HOW TO FIND IT**

Enable Fable 5.1 for all users in your enterprise org so they can run deep, context-heavy agentic sessions including software vulnerability discovery.

> 📍 In the AWS Management Console, go to Kiro settings, turn on the `Model availability` toggle under Model Governance, then select `Fable 5.1` from the model list. Once saved, Fable 5.1 appears in the model selector across all Kiro clients for every user.

Adds Claude Fable 5.1 to the model selector for Kiro enterprise customers, enabled via Model Governance in the AWS Management Console by turning on the `Model availability` toggle and selecting Fable 5.1 from the model list.

Fable 5.1 ships with a 1M context window and a 6x credit multiplier, supporting long-running agentic sessions and multi-file refactors at greater depth than prior models.

Fable 5.1 includes updated cybersecurity safeguards that allow software vulnerability discovery tasks, while exploit generation, binary-based vulnerability scanning, and penetration testing still redirect to Opus-class models.

`announced · Sep 14, 2026` `BLOG ↗`

GPT-5.6 Sol, Terra, and Luna now support 1M token context windows in Kiro IDE, CLI, and Web.

└──▷ HOW TO FIND IT

Access the expanded 1M-token GPT-5.6 models in Kiro Web after the rollout reaches your account.

> 📍 In the Kiro Web interface, refresh your browser, then open the model selector and choose GPT-5.6 Sol, Terra, or Luna to use the 1M token context window.

Extends GPT-5.6 Sol, Terra, and Luna context windows from 272K to 1M tokens across the IDE, CLI, and Web, enabling full-codebase ingestion in a single request without chunking or summarization.

Requests exceeding 272K tokens are billed at double the short-context credit multiplier (GPT-5.6 Sol 4.4x, Terra 2.2x, Luna 1.1x becomes 8.8x, 4.4x, 2.2x respectively) for Kiro Pro, Pro+, Pro Max, and Power customers.

WAS THIS USEFUL? 👍 👎

Zed

Feedback ⓘ announced · Sep 16, 2026 BLOG ↗ RANK

Zed is an open-source code editor that runs collaborative development workflows.

> Zed launched Delta, a public beta multiplayer environment for coding with agents that replaces pull requests with collaborative threads backed by a new Git-compatible versioning layer called DeltaDB.

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

01 **DeltaDB versioning layer extending Git** NEW 60

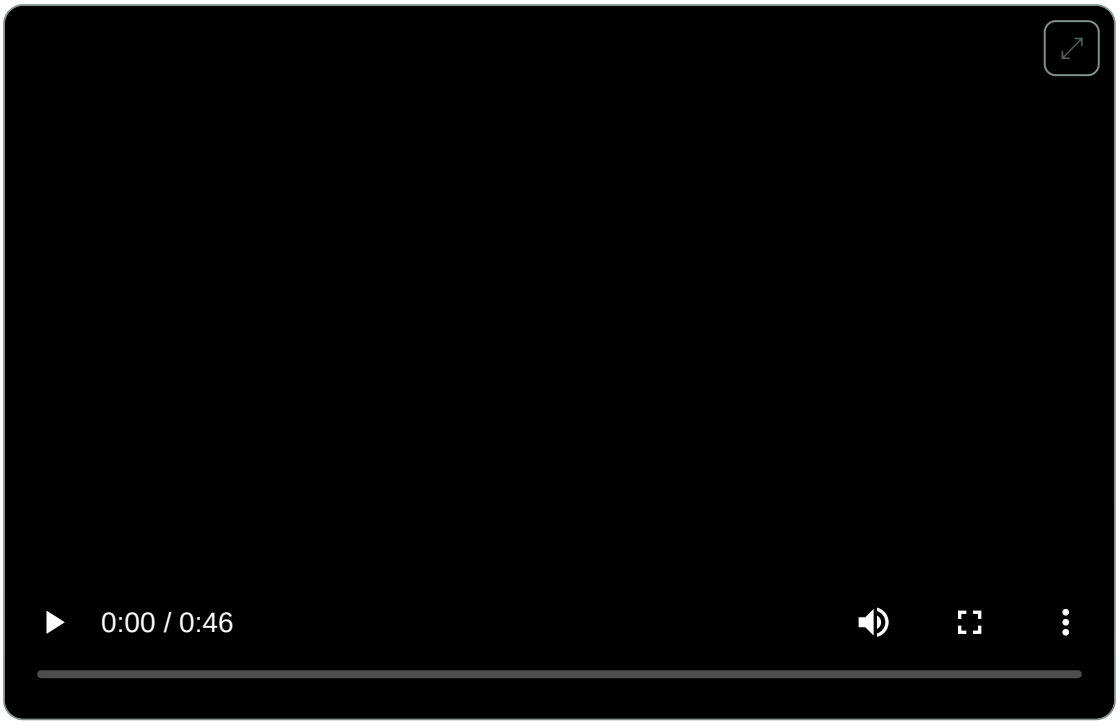
DeltaDB is a new versioning layer that extends Git, recording edits and agent/human messages between commits so it preserves how code evolved throughout a thread, while remaining compatible with existing Git repositories.

— Mechanism and compatibility explained, no API or config detail launch-20260916-419aeae3

THINNER COVERAGE BELOW

02 **Delta multiplayer agent-coding environment** NEW 55

Zed launched Delta, a new multiplayer environment for coding with agents and reviewing what they build. It is available as a download for macOS, Linux, and Windows, as a web app, and on mobile browsers.



— Names platforms but gives no setup or install steps launch-20260916-419aeae3

03 **Delta threads replace pull requests** NEW 50

Delta threads replace pull requests: teammates can join live agent conversations, share worktrees, and continue work across sessions without committing or pushing code.
.

— Mechanism described but no exact UI path or command launch-20260916-419aeae3

04 **Review subthreads for agent-assisted code review** NEW

50

Review subthreads are isolated copies of a parent thread's worktrees that let reviewers use agents to explore code and request revisions or apply fixes before the change lands.

— Explains behaviour but no concrete command to invoke it launch-20260916-419aeae3

> announced · Sep 16, 2026 BLOG ↗
>
> Zed launches Delta public beta: a multiplayer agent-coding environment that replaces pull requests with collaborative threads
>
> Launches Delta, a new multiplayer environment for coding with agents and reviewing what they build, available as a download for macOS, Linux, and Windows, as a web app, and on mobile browsers.
>
> Replaces pull requests with Delta threads, where teammates can join live agent conversations, share worktrees, and continue work across sessions without committing or pushing code.
>
> Adds review subthreads: isolated copies of a parent thread's worktrees that let reviewers use agents to explore code and request revisions or apply fixes before the change lands.
>
> Built on DeltaDB, a new versioning layer that extends Git by recording edits and agent/human messages between commits, preserving how code evolved throughout a thread while remaining compatible with existing Git repositories.

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.3.1 NOTES ↗

RANK

Zeroshot orchestrates executor-verifier agent loops that implement and validate software changes while logging workflow state to SQLite.

> Zeroshot v10.3.1 adds support for Codex 0.155.0's new browser, computer, and approval controls, updates its bundled agent CLI versions, and ships Python SDK wheels, while flagging a breaking change to Codex's approval policy setting.

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

01 **Codex retires untrusted approval policy value** BREAKING

70

Codex 0.155.0 rejects the retired `approval_policy="untrusted"` setting. Users who explicitly configured that value must manually update their Codex configuration before upgrading, since this release does not rewrite it automatically.

— Names exact config value and required manual migration step v10.3.1

THINNER COVERAGE BELOW

02 **Bundled agent CLI versions updated** IMPROVED

45

The target image now bundles updated agent CLIs: Claude Code 2.1.276, Codex 0.155.0, and GitHub Copilot 1.0.86.

— Exact versions named but no behavioral detail v10.3.1

03 **Codex 0.155.0 browser, computer, and approval controls** NEW

40

Zeroshot now recognizes Codex 0.155.0's new browser, computer, and approval controls and preserves explicitly configured policy for these settings.

— Names the controls but not mechanism of preservation v10.3.1

04 **Python SDK 10.3.1.post1 wheels** NEW

40

Python SDK 10.3.1.post1 wheels are now available in the SDK GitHub release.

— Concrete version and location but no usage detail v10.3.1

└──▷ **BREAKING ON UPGRADE**

! Codex 0.155.0 rejects the retired `approval_policy="untrusted"` setting — users who explicitly configured that value must update their Codex configuration manually before upgrading, as this release does not rewrite it automatically.

v10.3.1 NOTES ↗ LATEST STABLE

Zeroshot v10.3.1 adds Codex browser, computer, and approval controls and improves unavailable-target error reporting.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v10.3.1 https://github.com/the-open-engine/zeroshot.git
# already have the repo? check out this version:
$ git checkout v10.3.1
```

Recognizes Codex 0.155.0's new browser, computer, and approval controls and preserves explicitly configured policy.

Updates bundled agent CLIs in the target image: Claude Code 2.1.276, Codex 0.155.0, and GitHub Copilot 1.0.86.

Python SDK 10.3.1.post1 wheels are available in the SDK GitHub release.

└──▷ BREAKING ON UPGRADE

! Codex 0.155.0 rejects the retired `approval_policy="untrusted"` setting — users who explicitly configured that value must update their Codex configuration manually before upgrading, as this release does not rewrite it automatically.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

PydanticAI

Feedback ⓘ v2.45.0 NOTES ↗ RANK

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

> PydanticAI v2.45.0 adds a new model integration, TypeSafeModel, for TypeSafe's Jev.

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

01 **TypeSafeModel integration for TypeSafe's Jev** NEW 35

Adds `TypeSafeModel` for integration with TypeSafe's Jev.

— Names the class but no usage or mechanism given v2.45.0

v2.45.0 NOTES ↗ LATEST STABLE

PydanticAI v2.45.0 adds TypeSafeModel for TypeSafe's Jev integration.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.45.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.45.0
```

WAS THIS USEFUL? 👍 👎

Daily Pipecat

Feedback ⓘ v1.11.0 NOTES ↗ RANK

Pipecat is a framework for realtime voice and multimodal agents, orchestrating speech-to-text, model inference, and text-to-speech as a streaming pipeline.

> Pipecat's biggest addition is Pipecat Enterprise, letting teams run Pipecat Cloud agents inside their own Kubernetes cluster with a self-hosted control-plane/data-plane split, alongside v1.11.0's LiveKit video output, Gemini 3.8 Live support, a wave of new STT/TTS/LLM settings, and a much richer eval harness.

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

01 **Video output in LiveKit transport** NEW 90

Adds `video_out_enabled`, `video_out_codec`, `LiveKitParams.video_out_max_bitrate`, and `video_out_framerate` to `LiveKitTransport`, publishing a camera track from `RGB`, `RGBA`, `BGRA`, or `ARGB` frames.

Publish a video camera track from a LiveKit bot — useful when your agent needs to stream video alongside audio.

python

```python
from pipecat.transports.services.livekit import LiveKitTransport,
LiveKitParams

transport = LiveKitTransport(
    url=LIVEKIT_URL,
    token=LIVEKIT_TOKEN,
    room_name=ROOM_NAME,
    params=LiveKitParams(
        audio_out_enabled=True,
        video_out_enabled=True,
        video_out_codec='vp8',
        video_out_max_bitrate=1_000_000,
        video_out_framerate=30,
    ),
)
```

— Runnable code example with exact params and frame formats v1.11.0

02 **Eval harness scenario and manifest enhancements** NEW 90

Adds `text_excludes:` to scripted eval scenario expectations — the mirror of `text_contains:` — failing with kind `text_present` when matched text contains the

given substring; `concurrency:` on a manifest entry, capping parallel runs of that bot within the suite's overall concurrency; `name:` on a manifest entry, used as the display label, in the `-p` filter, in `results.jsonl` (a new `name` field beside `bot`), and in artifact file names; `data:` on a manifest entry's `runner_body:`, allowing the bot's runner-args body to be written inline as YAML or JSON; `EvalScriptTurnResult.expectations`, providing one `EvalExpectationResult` per expectation the turn resolved, recorded in `results.jsonl`; EvalScenarioFile.load() and a `scenarios:` top-level key allowing multiple named scenarios — scripted and simulated — to share one file with top-level defaults including `turns:` and `persona:` blocks; an `llm_marker` event and `marker:` field (`complete`, `short`, `long`, `incomplete`) for asserting on the bot LLM's turn-completion marker, with the harness event gaining `raw`, `markers`, and support for `marker_first`, `markers`, and `text_after` assertions; `eval:` on `function_call` scripted eval expectations, judging matched calls by name and arguments under a per-call judge prompt via the new EvalJudge.evaluate_call(); and `speed` in the harness's `speech:` block, setting Kokoro's rate multiplier for synthesized user audio (cache-keyed by speed).

Assert that a bot does NOT leak its internal turn-completion marker into spoken replies during eval.

```yaml
turns:
  - user: 'What time is it?'
    expect:
      - event: response
        text_excludes: '<complete>'
```

Group related short-answer and cutoff eval cases in one file to reduce boilerplate and run them together.

```yaml
name: turn_completion
judge: !include ../judge_text.yaml

scenarios:
  - name: short_answer
    turns:
      - user: 'Japan.'
        expect:
          - event: response
  - name: cutoff
    turns:
      - user: 'I would go to Japan because'
        expect:
          - event: response
            absent: true
            within_ms: 3000
```

— Every eval key named, with runnable YAML examples `v1.11.0`

03 **STT service settings and language expansions** NEW 85

Adds `partial_results_stability` to `AWSTranscribeSTTService.Settings`, selecting AWS Transcribe's `'high'`, `'medium'`, or `'low'` interim-result stability (default `'high'`); a `delay` field to `OpenAIRealtimeSTTService.Settings` for latency-versus-accuracy control (`minimal`, `low`, `medium`, `high`, `xhigh`), supported by `gpt-realtime-whisper`; a `keywords` field to `OpenAISTTService.Settings` for passing product names, acronyms, and specialized terms as transcription hints, supported by `gpt-transcribe`; `initial_prompt` to `WhisperSTTService.Settings`, giving Faster Whisper preceding context to steer transcript style, punctuation, and spelling; `SarvamRealtimeSTTService` support for Sarvam's `saaras:v4` second-generation realtime model (default remains `saaras:v3-realtime`); and `AssemblyAISTTService` `language_codes` steering gaining Urdu, Russian, Korean, Catalan, Galician, Romanian, Estonian, Persian, Cantonese, Afrikaans, Marathi, Zulu, Xhosa, and Norwegian Nynorsk, covering the full `universal-3-5-pro` and `universal-3-6-pro` language set.

— Every field, class and accepted value named exactly `v1.11.0`

04 **Pipecat Enterprise self-hosted Kubernetes deployment** NEW 83

Introduces Pipecat Enterprise: run Pipecat Cloud agents in your own Kubernetes cluster with the Pipecat Cloud control plane managing orchestration while your data plane handles agent workloads inside your VPC. Setup is Helm-based, covering the workloads namespace, external broker and log store, WebSocket exposure, replica counts, and pull

secrets via a region package values reference; Kubernetes Secrets stored in your cluster can be referenced as Pipecat Cloud secret sets for `env` and `image-pull` secrets. Agent sizing is set via CPU/memory values or organization agent profiles, targeting `amd64` or `arm64` node pools, and WebSocket endpoints for telephony and client sessions are exposed via Kubernetes Ingress or Gateway API with token auth. An optional log store captures agent logs and session metrics (requiring `cgroup v2` and specific base images), and reference architectures cover Amazon EKS and Oracle OKE, including VPC/VCN layout, node pools, managed broker and log store, and WebSocket front door.

— Extensive mechanism named but no runnable setup command given `product docs`

---

05 **LLM service model and reasoning support** NEW 82

Adds `reasoning_effort` to `GroqLLMService.Settings`, controlling reasoning effort on GPT-OSS and Qwen models (`'low'`, `'medium'`, `'high'` or `'none'`, `'default'` depending on model; `'none'` also strips `<think>` tokens from spoken responses); `GeminiLiveLLMService` support for `gemini-3.8-live` and `gemini-3.8-live-extended-thinking`, including `NON_BLOCKING` function calling by default, `BLOCKING` tools on `gemini-3.8-live`, and `thinking_level` (defaulting to `LOW`) on the extended-thinking model; a `mode` field on OpenAI Responses `ReasoningConfig` selecting `standard` or `pro` reasoning mode on supporting models such as `gpt-5.6`; and `SarvamLLMService` support for `deepseekv4-flash`, DeepSeek V4 Flash with a 1M-token context window, tool calling, and reasoning.

— Exact model names and config fields across four LLM services `v1.11.0`

---

06 **pipecat init scaffolding with Pipecat UI** IMPROVED 78

`pipecat init` React clients (Vite and Next.js) now scaffold using Pipecat UI, with components installed as source under `src/components/pipecat`, `components.json` pointing at the `@pipecat` registry, and `npx shadcn@latest add @pipecat/<component>` for adding more.

— Exact paths and runnable install command given `v1.11.0`

---

07 **TTS service settings additions** NEW 75

Adds an `effect` setting to `AzureTTSService` and `AzureHttpTTSService`, passing Azure's `<voice effect>` audio effect processor (`eq_car`, `eq_telecomhp8k`) to the synthesis request; `math_notation` to `SmallestTTSService.Settings`, enabling Smallest's opt-in flag for reading digit-flanked math operators as words (e.g. `2 + 2` as `two plus two`); and `client_reference_id` to `SonioxTTSService` settings, the identifier Soniox records with each request in its usage logs.

— Named fields and services but no runnable example `v1.11.0`

---

08 **Turn-completion marker reporting via RTVI** NEW 71

Adds `bot_llm_marker_enabled` parameter to `RTVIObserver` (off by default) and a `bot-llm-marker` RTVI message reporting the bot LLM's turn-completion marker (`complete`, `short`, or `long`); adds `LLMMarkerResponseFrame`, pushed by `UserTurnCompletionLLMServiceMixin`, carrying the `raw` text, the marker read, and the `markers` recognized.

— Exact params and frame named but no usage example `v1.11.0`

---

09 **Pipecat Enterprise region CLI and diagnostics** NEW

67

Adds the `pipecat cloud regions` CLI subcommand to list available deployment regions and manage the lifecycle of self-hosted Enterprise regions, plus Diagnostics tooling including a `doctor` command, a region status and event feed, and a support bundle.

List available cloud regions or check the status of your self-hosted Enterprise region from the CLI.

```
$ pipecat cloud regions
```

— Names exact runnable CLI commands `product docs`

THINNER COVERAGE BELOW

10 **Audio resampler stream-boundary control** NEW

53

Adds BaseAudioResampler.flush() and BaseAudioResampler.reset() for explicit stream-boundary control: flush() returns buffered audio for a completed stream, reset() discards it for an abandoned one.

— Exact method names with behaviour, no usage example `v1.11.0`

---

11 **HeyGen LiveAvatar session duration cap** NEW

42

Adds `max_session_duration` to `LiveAvatarNewSessionRequest`, capping a HeyGen LiveAvatar session at a given number of seconds.

— Named field but thin, one-line description `v1.11.0`

---

12 **Oruk speech-to-text service** NEW

40

Adds the Oruk Speech-to-Text service with realtime speech and phrase estimates, plus optional local Orukeet CPU transcription for completed utterances.

— Describes capability but no service class or config named `product docs`

---

13 **pipecat-effects audio filter service** NEW

30

Adds the `pipecat-effects` audio filter service for applying audio effects to bot output.

— Single-line addition with no mechanism detail `product docs`

v1.11.0 NOTES ↗ LATEST STABLE

Pipecat v1.11.0 adds video output to LiveKit, Gemini 3.8 Live support, new STT/TTS settings, and a richer eval harness with multi-scenario files.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.11.0 https://github.com/pipecat-ai/pipecat.git
# already have the repo? check out this version:
$ git checkout v1.11.0
```

└──▷ **USE IT**

Publish a video camera track from a LiveKit bot — useful when your agent needs to stream video alongside audio.

python

```python
from pipecat.transports.services.livekit import LiveKitTransport, LiveKitParams

transport = LiveKitTransport(
    url=LIVEKIT_URL,
    token=LIVEKIT_TOKEN,
    room_name=ROOM_NAME,
    params=LiveKitParams(
        audio_out_enabled=True,
        video_out_enabled=True,
        video_out_codec='vp8',
        video_out_max_bitrate=1_000_000,
        video_out_framerate=30,
    ),
)
```

Assert that a bot does NOT leak its internal turn-completion marker into spoken replies during eval.

yaml

```yaml
turns:
  - user: 'What time is it?'
    expect:
      - event: response
        text_excludes: '<complete>'
```

Group related short-answer and cutoff eval cases in one file to reduce boilerplate and run them together.

```yaml
name: turn_completion
judge: !include ../judge_text.yaml

scenarios:
  - name: short_answer
    turns:
      - user: 'Japan.'
        expect:
          - event: response
  - name: cutoff
    turns:
      - user: 'I would go to Japan because'
        expect:
          - event: response
            absent: true
            within_ms: 3000
```

Adds `video_out_enabled`, `video_out_codec`, `LiveKitParams.video_out_max_bitrate`, and `video_out_framerate` to `LiveKitTransport`, publishing a camera track from `RGB`, `RGBA`, `BGRA`, or `ARGB` frames.

Adds `partial_results_stability` to `AWSTranscribeSTTService.Settings`, selecting AWS Transcribe's `'high'`, `'medium'`, or `'low'` interim-result stability (default: `'high'`).

Adds an `effect` setting to `AzureTTSService` and `AzureHttpTTSService`, passing Azure's `<voice effect>` audio effect processor (`eq_car`, `eq_telecomhp8k`) to the synthesis request.

Adds `reasoning_effort` to `GroqLLMService.Settings`, controlling reasoning effort on GPT-OSS and Qwen models; accepted values are `'low'`, `'medium'`, `'high'` or `'none'`, `'default'` depending on model, and `'none'` also strips `<think>` tokens from spoken responses.

Adds a `delay` field to `OpenAIRealtimeSTTService.Settings` for latency-versus-accuracy control (`minimal`, `low`, `medium`, `high`, `xhigh`), supported by `gpt-realtime-whisper`.

+23 more

WAS THIS USEFUL?  

Cloudflare Agents

Feedback ⓘ `@cloudflare/think@0.19.0`

NOTES ↗ RANK

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

> Cloudflare Agents shipped a non-React WebSocket transport for AI SDK chat UIs and a substantial rework of the queue lifecycle system, including async queue APIs, new lifecycle status semantics, and breaking storage migrations.

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

01 Queue lifecycle rework in agents/queue

BREAKING

90

Adds a Queue Lifecycle capability (`agents/queue`) for durable, alarm-driven background work with typed callbacks, `options.id` support on queue() for stable item identity, and per-item retry. `dequeue` , `dequeueAll` , `dequeueAllByCallback` , `getQueue` , and `getQueues` are now asynchronous — callers that did not await them will silently drop results — and `QueueItem.created_at` is renamed `createdAt` . LifecycleServices.starting() is removed and replaced by status(), returning `'zero' | 'starting' | 'started'` ; calling starting() now throws. Queued callbacks run from the alarm loop no longer see the enqueuing request's `connection` or `request` through getCurrentAgent(). The `cf_agents_queues` table is removed, with legacy rows migrating into the job queue on next start, and the `cf_think_workflow_notifications` table migrates and is dropped on start — deployments skipping this release must upgrade through it before the next minor release removes the migration path. Workflow-notification delivery now retries with backoff capped at ten minutes (previously five) and gives up after twelve hours of continuous failure (previously never), reporting the failure through `onError` . `agents >=0.24.0` is now required for Think's background work.

— Detailed mechanism and migration notes but no runnable code example

`@cloudflare/think@0.19.0`

02 Non-React WebSocket chat transport for AI SDK UIs

NEW

70

Adds `WebSocketChatTransport` from `agents/chat/transport` to connect AI SDK chat UIs in any non-React framework without requiring React as a peer dependency, and exports `AgentConnection` connection types from the same module for non-React client use. Usage shown includes wiring a vanilla JS client with new WebSocketChatTransport({ host: window.location.host }) and a Vue component using `useChat` from `@ai-sdk/vue` with `AgentClient` from `agents/client` .

Wire up an AI SDK chat UI in a vanilla JS or non-React framework using a WebSocket connection to an Agents backend.

```javascript
import { WebSocketChatTransport } from 'agents/chat/transport';

const transport = new WebSocketChatTransport({
  host: window.location.host,
});
```

Wire up a Vue chat component to a Cloudflare Agent without pulling in React, using `WebSocketChatTransport` as the AI SDK transport layer.

```javascript
import { useChat } from "@ai-sdk/vue";
import { AgentClient } from "agents/client";
import { WebSocketChatTransport } from "agents/chat/transport";

const client = new AgentClient({
  agent: "<your-agent>",
  name: "user-123",
  host: window.location.host,
});

const { messages, input, handleSubmit } = useChat({
  transport: new WebSocketChatTransport(client),
  cancelOnClientAbort: true,
});
```

— Names module and class with runnable code, but little mechanism beyond usage

product docs

└──▷ BREAKING ON UPGRADE

! `dequeue`, `dequeueAll`, `dequeueAllByCallback`, `getQueue`, and `getQueues` are now asynchronous — callers that did not await them will silently drop results.

! `QueueItem.created_at` is renamed `createdAt` — any code reading `created_at` will see `undefined` after upgrade.

! LifecycleServices.starting() is removed and replaced by status() — any call to starting() will throw after upgrade.

! Queued callbacks running from the alarm loop no longer see the enqueuing request's `connection` or `request` through getCurrentAgent() — code that relied on those values inside a queued callback will find them absent.

! The `cf_agents_queues` table is gone; legacy rows migrate into the job queue on next start. Deployments skipping this release must upgrade through it before the next minor release

removes the migration path.

! The `cf_think_workflow_notifications` table migrates and is dropped on start; deployments skipping this release must upgrade through it before the migration is removed in the next minor release.

! `agents >=0.24.0` is now required for Think's background work.

`@cloudflare/think@0.19.0` NOTES ↗ LATEST STABLE

Adds Queue Lifecycle capability (`agents/queue`) for durable, alarm-driven background work with typed callbacks, upsert IDs, and per-item retry.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @cloudflare/think@0.19.0
https://github.com/cloudflare/agents.git
# already have the repo? check out this version:
$ git checkout @cloudflare/think@0.19.0
```

Makes `dequeue` , `dequeueAll` , `dequeueAllByCallback` , `getQueue` , and `getQueues` asynchronous, and adds `options.id` support to queue() for stable item identity.

Replaces LifecycleServices.starting() with status(), which returns `'zero' | 'starting' | 'started'` .

Workflow-notification delivery now retries with backoff capped at ten minutes (previously five) and gives up after twelve hours of continuous failure (previously never), reporting the failure through `onError` .

└──▷ **BREAKING ON UPGRADE**

! `dequeue` , `dequeueAll` , `dequeueAllByCallback` , `getQueue` , and `getQueues` are now asynchronous — callers that did not await them will silently drop results.

! `QueueItem.created_at` is renamed `createdAt` — any code reading `created_at` will see `undefined` after upgrade.

! LifecycleServices.starting() is removed and replaced by status() — any call to starting() will throw after upgrade.

! Queued callbacks running from the alarm loop no longer see the enqueuing request's `connection` or `request` through getCurrentAgent() — code that relied on those values inside a queued callback will find them absent.

! The `cf_agents_queues` table is gone; legacy rows migrate into the job queue on next start. Deployments skipping this release must upgrade through it before the next minor release removes the migration path.

! The `cf_think_workflow_notifications` table migrates and is dropped on start; deployments skipping this release must upgrade through it before the migration is removed in the next minor release.

! `agents >=0.24.0` is now required for Think's background work.

WAS THIS USEFUL? 👍 👎

Mem0 Feedback ⓘ ts-v3.2.0 NOTES ↗ RANK

Mem0 is a memory layer for AI agents that extracts, stores, and retrieves durable facts across sessions through a hosted or self-hosted store.

> Mem0 shipped a chat-based Copilot for inspecting and tuning memory extraction from the dashboard, plus a client-identity header system that lets the platform attribute calls across nested SDK layers.

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

01 **Copilot AI assistant in dashboard** NEW 85

Adds Copilot, an AI assistant embedded in the Mem0 dashboard sidebar that can inspect stored memories, suggest extraction rules and categories, and change project settings through natural-language chat. Review changes mode pauses before writing any project configuration or test memory for approval, while Auto-apply mode writes settings and test memories without approval after confirming the selected project and scope. A Scope selector below the message box filters inspection and suggestions by User, Agent, App, or Run ID. Users can add a test memory for a specific user directly from chat, returning an `event_id` checkable via the Get Event API to track extraction status (`PENDING`, `SUCCEEDED`, `FAILED`). Copilot also suggests Custom categories from stored memories and Custom extraction instructions by comparing recent add requests with extracted memories, and exposes Memory depth (Less, Medium, More detailed) and Multilingual behavior as chat-adjustable project settings. A History panel shows the 50 most recently active chats for the selected project with New chat and Delete chat actions, and proposal cards support inline Edit with Apply with edits and Discard edits.

— Detailed UI mechanism and named controls, but no CLI/API to run directly product docs

02 **Client identity headers across SDK layers** NEW 85

Adds X-Mem0-Source, X-Application, and X-Mem0-Client headers to all client requests so the platform can identify which product and SDK layer chain made each call. X-Mem0-Source and X-Application are sent as set-once headers preserving the outer wrapper's identity, while X-Mem0-Client is append-only and reports the full layer chain, bounded by dropping whole `name/version` entries rather than slicing characters. The TypeScript SDK (v3.2.0) reads `MEM0_SOURCE`, `MEM0_APPLICATION`, and `MEM0_CLIENT_STACK` environment variables to populate these headers, matching the Python SDK's behavior, and injects the SDK version into X-Mem0-Client at build time so it never goes stale.

Identify a plugin wrapper in a multi-layer stack so the platform attributes calls to the correct surface and host app — without modifying code that already set its own identity.

```
$ export MEM0_SOURCE=my-plugin
export MEM0_APPLICATION=my-host-app
export MEM0_CLIENT_STACK=my-plugin/1.0.0
```

— Named headers and env vars with a runnable export example ts-v3.2.0 v2.1.0

ts-v3.2.0 NOTES ↗ LATEST STABLE

Mem0 TypeScript SDK now propagates `X-Mem0-Source`, `X-Application`, and `X-Mem0-Client` headers, readable via environment variables.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch ts-v3.2.0 https://github.com/mem0ai/mem0.git
# already have the repo? check out this version:
$ git checkout ts-v3.2.0
```

Reads `MEM0_SOURCE`, `MEM0_APPLICATION`, and `MEM0_CLIENT_STACK` environment variables to populate the `X-Mem0-Source`, `X-Application`, and `X-Mem0-Client` request headers, matching the Python SDK's behavior.

Sends `X-Mem0-Source` and `X-Application` as set-once headers (outer wrapper identity is preserved) and `X-Mem0-Client` as an append-only header that reports the full layer chain.

Injects the SDK version into `X-Mem0-Client` at build time so the reported version never goes stale across releases.

v2.1.0 NOTES ↗ STABLE

Mem0 v2.1.0 adds three client identity headers and environment variables for surface-identity propagation across SDK layers.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.1.0 https://github.com/mem0ai/mem0.git
# already have the repo? check out this version:
$ git checkout v2.1.0
```

└──▷ TRY IT

Identify a plugin wrapper in a multi-layer stack so the platform attributes calls to the correct surface and host app — without modifying code that already set its own identity.

```
$ export MEM0_SOURCE=my-plugin
export MEM0_APPLICATION=my-host-app
export MEM0_CLIENT_STACK=my-plugin/1.0.0
```

Adds `X-Mem0-Source`, `X-Application`, and `X-Mem0-Client` headers to all client requests so the platform can identify which product and SDK layer chain made each call.

Makes `X-Mem0-Client` append-only and bounded by dropping whole `name/version` entries rather than slicing characters, preserving valid identifiers across the full call chain.

WAS THIS USEFUL? 👍 👎

Microsoft Agent Framework

Feedback ⓘ python-1.19.0

NOTES ↗ RANK

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

> Microsoft Agent Framework's python-1.19.0 release adds new vector-store connectors for MongoDB, Azure DocumentDB, and Cosmos DB alongside tool-exposure and orchestration improvements, and ships four breaking changes touching HTTP cookie handling, MCP sessions/archives, and Redis key scoping.

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

01 **Vector-store connectors for MongoDB, Azure DocumentDB, and Cosmos DB**

NEW

65

Adds the alpha `agent-framework-azure-documentdb` Azure DocumentDB vector-store connector, an Azure Cosmos DB NoSQL implementation of the shared vector-store APIs via `agent-framework-azure-cosmos`, and a MongoDB vector-store connector, alongside new generic vector-store provider protocols added to `agent-framework-core`.

— Names three connector packages but no usage example given python-1.19.0

02 **Explicit HTTP cookie persistence for HTTP clients**

BREAKING

65

HTTP cookie persistence is now explicit for framework-owned and caller-owned clients across `agent-framework-a2a`, `agent-framework-ag-ui`, `agent-framework-core`, and `agent-framework-declarative` — implicit cookie carry-over no longer works.

— Names four affected packages and the behavior change clearly python-1.19.0

03 **Redis history storage key scoping**

BREAKING

60

Redis history storage keys in `agent-framework-redis` (beta) are now scoped by provider and session identity — existing keys stored under the old scheme will not be found.

— Explains before/after and impact on existing stored keys python-1.19.0

THINNER COVERAGE BELOW

04 **Tool exposure controls: AgentModeProvider and CodeAct schemas**

NEW

50

Adds per-tool `AgentModeProvider` exposure controls to `agent-framework-core`, and exposes `CodeAct` tool parameter schemas with configurable compact or JSON descriptions across `agent-framework-core`, `agent-framework-hyperlight`, and `agent-framework-monty`.

— Named surfaces but mechanism of exposure control unexplained python-1.19.0

05 **Aspire trace display in DevUI** NEW

45

Displays Aspire traces in the `agent-framework-devui` developer UI.

— Names the UI package but not navigation details python-1.19.0

06 **Stable workflow names and checkpoint restoration in orchestrations**

IMPROVED

45

Adds stable names for built-in orchestration workflows and registers their checkpoint types for restoration in `agent-framework-orchestrations`.

— Explains behavior change but no API or config example python-1.19.0

07 **MCP skill archives restricted to ZIP format** BREAKING

45

MCP skill archives are now limited to ZIP files only; non-ZIP archive formats will be rejected.

— Clear behavior change but no migration guidance given python-1.19.0

08 **Per-invocation scoping for provider-backed MCP sessions**

45

BREAKING

Provider-backed MCP sessions in `agent-framework-declarative` are now scoped per invocation rather than shared.

— States the scoping change but not its practical impact python-1.19.0

09 **Sequential function-call invocation option** NEW

35

Adds an option to invoke function calls sequentially in `agent-framework-core`.

— No flag name or config key given for the option python-1.19.0

10 **Instrumentation message-event controls** NEW

25

Adds instrumentation message-event controls to `agent-framework-core`.

— Bare mention with no detail on controls' scope or use python-1.19.0

11 **MCP sampling callback deprecated** DEPRECATED 25

The MCP sampling callback is deprecated.

— Only names the deprecated callback with no replacement noted python-1.19.0

└──▷ **BREAKING ON UPGRADE**

! HTTP cookie persistence is now explicit for framework-owned and caller-owned clients across `agent-framework-a2a`, `agent-framework-ag-ui`, `agent-framework-core`, and `agent-framework-declarative` — implicit cookie carry-over will no longer work.

! MCP skill archives are now limited to ZIP files only; non-ZIP archive formats will be rejected, and the MCP sampling callback is deprecated.

! Provider-backed MCP sessions in `agent-framework-declarative` are now scoped per invocation rather than shared.

! Redis history storage keys in `agent-framework-redis` (beta) are now scoped by provider and session identity — existing keys stored under the old scheme will not be found.

python-1.19.0 NOTES ↗ LATEST STABLE

Adds MongoDB, Azure DocumentDB, and Cosmos DB vector-store connectors, sequential function-call invocation, Aspire trace display in DevUI, and CodeAct tool schema exposure.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch python-1.19.0
https://github.com/microsoft/agent-framework.git
# already have the repo? check out this version:
$ git checkout python-1.19.0
```

Adds the alpha `agent-framework-azure-documentdb` Azure DocumentDB vector-store connector.

Adds an Azure Cosmos DB NoSQL implementation of the shared vector-store APIs via `agent-framework-azure-cosmos`.

Adds generic vector-store provider protocols and per-tool `AgentModeProvider` exposure controls to `agent-framework-core`.

Adds an option to invoke function calls sequentially in `agent-framework-core`.

Exposes `CodeAct` tool parameter schemas with configurable compact or JSON descriptions across `agent-framework-core`, `agent-framework-hyperlight`, and `agent-framework-monty`.

+3 more

└──▷ **BREAKING ON UPGRADE**

- ! HTTP cookie persistence is now explicit for framework-owned and caller-owned clients across `agent-framework-a2a`, `agent-framework-ag-ui`, `agent-framework-core`, and `agent-framework-declarative` — implicit cookie carry-over will no longer work.
- ! MCP skill archives are now limited to ZIP files only; non-ZIP archive formats will be rejected, and the MCP sampling callback is deprecated.
- ! Provider-backed MCP sessions in `agent-framework-declarative` are now scoped per invocation rather than shared.
- ! Redis history storage keys in `agent-framework-redis` (beta) are now scoped by provider and session identity — existing keys stored under the old scheme will not be found.

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

gram

Feedback ⓘ server@2.7.0 NOTES ↗ RANK

Gram provides MCP servers and AI-agent tools from APIs with access controls and observability.

> Gram's biggest addition this window is full Pi (pi.dev) agent observability — install, session capture, and policy enforcement — alongside a cluster of OAuth/session-security features (ID-JAG grants, tunneled OAuth back-channel, private_key_jwt, CIMD) and an expanded Shadow AI inventory and detection library; two breaking changes affect ChatGPT gateway blocking and trial administration.

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

01 Pi agent observability and session capture NEW

93

Adds `speakeasy-hooks install --provider=pi` to generate and install a local Pi (pi.dev) observability extension that forwards Pi lifecycle events over NDJSON stdio to the hooks relay; the relay reads `.pi/mcp.json` and `~/.pi/agent/mcp.json` to inventory MCP servers reachable from a Pi workspace and attribute tool calls to the matching server under the `pi` source. The dashboard captures Pi sessions — prompts, tool calls and results, per-turn tokens and cost, and session start/end — under the `pi` source, with policy deny decisions blocking prompts or tool calls inside Pi in real time.

Install the Pi observability extension into a repository's local Pi directory to capture prompts, tool calls, and session events in the Gram dashboard.

```
$ speakeasy-hooks install --provider=pi
```

Install the Pi observability extension into the current repository so gram can capture Pi sessions and enforce policy on them locally.

```
$ speakeasy-hooks install --provider=pi
```

— Exact install command, config file paths, and dashboard mechanism all named.

server@2.7.0 dashboard@0.121.0

02 Tunnel-routed OAuth back-channel for remote issuers NEW

90

Remote session issuers can now be bound to a tunneled MCP server so the full OAuth back-channel — code exchange, token refresh, revocation, dynamic client registration,

JWK Set, userinfo, and introspection — routes through the tunnel when the authorization server is unreachable; every issuer endpoint must be reachable from the same local origin as `TUNNEL_LOCAL_MCP_URL`.

— Names every routed endpoint type and the exact env var. `server@2.7.0`

03 **Shadow AI inventory and scan target library** NEW 85

Adds `list_shadow_ai_inventory` and `list_ai_scan_library` Platform MCP tools (org-admin required) to enumerate detected AI tools and the scan target library. The dashboard gains a Shadow AI section listing AI tools running on an organization's devices (coding harnesses, assistants, open model runners) with logo and access decision, plus a new AI scan targets page showing built-in targets grouped by kind and org-specific targets above them, with the ability to add a target by pasting JSON or fetching from a URL.

— Named MCP tools and exact dashboard pages/capabilities. `server@2.7.0` `dashboard@0.121.0`

04 **Watchdog risk analysis status endpoint and display** NEW 83

Adds the `risk.getAnalysisStatus` management API endpoint and `get_risk_analysis_status` Platform MCP tool to surface when Watchdog risk analysis last ran; the Watchdog dashboard page now shows that status as `Analyzing now`, `Last analyzed 4m ago`, or `No recent analysis`.

— Named endpoint, MCP tool, and exact status strings shown. `server@2.7.0` `dashboard@0.121.0`

05 **private_key_jwt authentication for OAuth token exchange** NEW 74

Adds `private_key_jwt` authentication for remote OAuth token exchanges and refreshes, letting remote session clients select an attached organization JSON Web Key Set and configure the assertion audience.

— Names exact auth method and configurable options. `server@2.7.0`

06 **Shadow AI detection for Pi and ChatGPT variants** IMPROVED 70

Shadow AI now detects Pi (pi.dev) on enrolled devices, matched by its `pi` binary, process name, and `~/.pi` configuration directory. It also distinguishes the current ChatGPT desktop app from ChatGPT Classic via separate bundle IDs, reporting which builds are installed and running per device, and now reports ChatGPT Classic as running (not just installed) by tracking its process name alongside its bundle ID.

— Names exact binaries, bundle IDs, and directories used for detection. `server@2.7.0`

07 **ID-JAG JWT bearer grants for session tokens** NEW 67

User session token endpoints configured with a trusted remote session issuer now accept ID-JAG JWT bearer grants, issuing short-lived, resource-bound access tokens without refresh tokens.

— Names grant type and token behavior, no exact endpoint given. server@2.7.0

08 **ChatGPT gateway matcher consolidation** BREAKING 67

The ChatGPT gateway matchers move from the ChatGPT Classic target onto the new ChatGPT target; organizations that blocked ChatGPT Classic to refuse ChatGPT traffic must now record that decision against ChatGPT, as Classic is no longer enforceable on its own.

— Explains the exact migration action operators must take. server@2.7.0

09 **MCP gateway matching by CIMD vendor key** IMPROVED 64

The MCP gateway now matches calling AI tools by CIMD vendor key, OAuth client ID, or client name reported at initialize, and enforces the organization's block decision at the gateway.

— Names matching criteria but no exact API surface given. server@2.7.0

10 **Change end date replaces trial extension action** BREAKING 60

Replaces the admin trial extension action with `Change end date`, allowing operators to shorten or extend a running trial to any future UTC calendar date, with an audit record.

— Names the exact replaced action and its new behavior. server@2.7.0

THINNER COVERAGE BELOW

11 **CIMD client authentication alternative** NEW 54

Supports CIMD Client ID Metadata Document as a stable authentication alternative to dynamic client registration for assistants connecting to CIMD-capable MCP OAuth servers.

— Names the standard but no configuration surface given. server@2.7.0

12 **Workload subject kind for session URNs** NEW 50

Adds a `workload` kind to the session subject URN so a machine vouched for by an external issuer can be the subject of a Gram-issued session.

— Names the URN kind but gives little further mechanism. server@2.7.0

└──▷ BREAKING ON UPGRADE

! The ChatGPT gateway matchers move from the ChatGPT Classic target onto the new ChatGPT target; organizations that blocked ChatGPT Classic to refuse ChatGPT traffic must now record that decision against ChatGPT, as Classic is no longer enforceable on its own.

server@2.7.0 NOTES ↗ LATEST STABLE

gram 2.7.0 adds Pi agent observability, ID-JAG JWT grants, tunnel-routed OAuth back-channel, and private_key_jwt client auth.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch server@2.7.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout server@2.7.0
```

└──▷ TRY IT

Install the Pi observability extension into a repository's local Pi directory to capture prompts, tool calls, and session events in the Gram dashboard.

```
$ speakeasy-hooks install --provider=pi
```

Adds `speakeasy-hooks install --provider=pi` to generate and install the Pi (pi.dev) observability extension locally, forwarding Pi lifecycle events over NDJSON stdio to the hooks relay.

Adds `risk.getAnalysisStatus` management API endpoint and `get_risk_analysis_status` Platform MCP tool to surface when Watchdog risk analysis last ran.

Adds `list_shadow_ai_inventory` and `list_ai_scan_library` Platform MCP tools (org-admin required) to enumerate detected AI tools and the scan target library.

Adds `private_key_jwt` authentication for remote OAuth token exchanges and refreshes, letting remote session clients select an attached organization JSON Web Key Set and configure the assertion audience.

Accepts ID-JAG JWT bearer grants at user session token endpoints configured with a trusted remote session issuer, issuing short-lived, resource-bound access tokens without refresh tokens.

+9 more

└──▷ BREAKING ON UPGRADE

! The ChatGPT gateway matchers move from the ChatGPT Classic target onto the new ChatGPT target; organizations that blocked ChatGPT Classic to refuse ChatGPT

traffic must now record that decision against ChatGPT, as Classic is no longer enforceable on its own.

`dashboard@0.121.0` NOTES ↗ STABLE

gram: Adds `.pi/mcp.json` and `~/.pi/agent/mcp.json` config-file reading so the relay inventories MCP servers reachable from a Pi workspace and attributes tool calls to the correct server for Shadow MCP visibility.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch dashboard@0.121.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout dashboard@0.121.0
```

└──▷ **TRY IT**

Install the Pi observability extension into the current repository so gram can capture Pi sessions and enforce policy on them locally.

```
$ speakeasy-hooks install --provider=pi
```

Captures Pi (pi.dev) sessions — prompts, tool calls and results, per-turn tokens and cost, session start/end — in the dashboard under the `pi` source, with policy deny blocking prompts or tool calls inside Pi in real time.

Adds a Shadow AI section to the dashboard listing AI tools running on an organization's devices (coding harnesses, assistants, and open model runners) with logo and access decision.

Adds a new AI scan targets page showing the detection library, with built-in targets grouped by kind, org-specific targets above them, and the ability to add a target by pasting JSON or fetching from a URL.

Adds Watchdog page status display showing when risk analysis last ran (`Analyzing now` , `Last analyzed 4m ago` , or `No recent analysis`).

WAS THIS USEFUL? 👍 👎

Composio

Feedback ⓘ announced · Sep 14, 2026 BLOG ↗ RANK

Composio connects AI agents to external tools and services with managed authentication and integrations.

> Composio's window centered on Shared Connections, letting a single app account be securely shared across an organization's agents, alongside a new HighLevel MCP toolkit integration.

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

01 **Shared Connections for team-wide app access** NEW 90

A `Connect for your team` option on the Apps page in the Composio dashboard lets an account owner create a team-wide connection to apps like Gmail, Meta Ads, or QuickBooks without distributing credentials. Personal connections always take priority over shared ones for a user who has their own account linked; teammates without a personal connection automatically fall back to the shared team connection. Team members can be invited to a shared connection by email directly from the connection view, or later via Settings → Members, with new invitees receiving an org invitation and gaining access to all shared connections on joining. The app dashboard's `Connected Accounts` section shows usage history — who used the connection and when — alongside options to invite more teammates or delete the connection. The feature is available as experimental for developers building on the Composio platform, letting agents serve company-managed accounts and shared mailboxes across multiple users.

Create a shared team connection to a company app so all organization members can use it in their agents without needing individual credentials.

📍 1. Log in to the Composio dashboard and go to the Apps page. 2. Find the app (e.g. Gmail, Meta Ads, QuickBooks). 3. Open the dropdown next to Connect and select `Connect for your team`. 4. Sign in with the shared account credentials to authorize the team connection. 5. The connection now appears under Connected Accounts and is available to all org members who lack a personal connection.

Invite a teammate to use existing shared connections so they can access all team-connected apps without connecting each account individually.

📍 1. In the Composio dashboard, open the app with an active shared connection. 2. Under Connected Accounts, select the shared connection and choose `Invite teammate`. 3. Enter the teammate's email address. 4. If the teammate is not yet in your org, they receive an invitation; once they join, all shared connections become available to them automatically. (Alternatively: go to Settings → Members to invite teammates after the connection is created.)

— Exact UI paths, invite flow, and fallback behaviour given launch-20260914-efd7196c

THINNER COVERAGE BELOW

02 **HighLevel MCP integration** NEW 35

Adds `HighLevel MCP` as a supported toolkit integration.

— Only the integration name is given, no detail product docs

announced · Sep 14, 2026 BLOG ↗

Composio adds Shared Connections so a single app account can be securely shared across an entire organization.

└──▷ **HOW TO FIND IT**

Create a shared team connection to a company app so all organization members can use it in their agents without needing individual credentials.

📍 1. Log in to the Composio dashboard and go to the Apps page. 2. Find the app (e.g. Gmail, Meta Ads, QuickBooks). 3. Open the dropdown next to Connect and select Connect for your team. 4. Sign in with the shared account credentials to authorize the team connection. 5. The connection now appears under Connected Accounts and is available to all org members who lack a personal connection.

Invite a teammate to use existing shared connections so they can access all team-connected apps without connecting each account individually.

📍 1. In the Composio dashboard, open the app with an active shared connection. 2. Under Connected Accounts, select the shared connection and choose Invite teammate. 3. Enter the teammate's email address. 4. If the teammate is not yet in your org, they receive an invitation; once they join, all shared connections become available to them automatically. (Alternatively: go to Settings → Members to invite teammates after the connection is created.)

Adds a Connect for your team option on the Apps page in the Composio dashboard, letting an account owner create a team-wide connection to apps like Gmail, Meta Ads, or QuickBooks without distributing credentials.

Personal connections always take priority over shared ones when a user has their own account linked; teammates without a personal connection automatically fall back to the shared team connection.

Team members can be invited to a shared connection by email directly from the connection view, or later via Settings → Members; new invitees receive an org invitation and gain access to all shared connections on joining.

Shared connections surface a Connected Accounts section in the app dashboard showing usage history — who used the connection and when — alongside options to

> invite more teammates or delete the connection.
>
> Shared Connections are available to developers building on the Composio platform as an experimental feature, allowing agents to serve company-managed accounts and shared mailboxes across multiple users.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Perplexity API

Feedback ⓘ snapshot-20260918 NOTES ↗ RANK

Perplexity API Platform - build with the Router, Agent, Search, and Embeddings APIs. Real-time, web-wide research and Q&A capabilities for your products.

> Perplexity API shipped a new gateway responses endpoint with detailed token accounting and agentic controls, plus MCP connector support and a new model option in the Agent API.

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

01 **Gateway responses endpoint in Router API** NEW 95

Adds `POST https://api.perplexity.ai/router/v1/responses` for submitting and retrieving gateway responses, returning structured `output`, `status`, and `usage` objects. Exposes per-request token accounting via `usage.input_tokens`, `usage.output_tokens`, `usage.total_tokens`, `input_tokens_details.cached_tokens`, `input_tokens_details.cache_write_tokens`, and `output_tokens_details.reasoning_tokens`. Supports `prompt_cache_key` and `store` for caching and persistence control; `parallel_tool_calls`, `tool_choice`, and `max_tool_calls` for agentic tool-use behaviour; `background` for async processing; `previous_response_id` for chaining multi-turn responses; `safety_identifier` for tagging responses with a caller-supplied safety context; and `service_tier`, `truncation`, `reasoning`, `top_logprobs`, `presence_penalty`, `frequency_penalty`, `top_p`, and `temperature` for model-routing and sampling control. The response body also exposes `incomplete_details.reason` to explain why a response did not fully complete.

Submit a gateway request and inspect token usage including cache hits and reasoning tokens for cost auditing.

```
$ curl --request POST \
  --url https://api.perplexity.ai/router/v1/responses \
  --header 'Authorization: Bearer <token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "instructions": "Summarise the threat report.",
    "store": true,
    "prompt_cache_key": "threat-report-v1",
    "temperature": 0.2,
    "max_output_tokens": 512
  }'
```

Chain a follow-up query to a previous response to build a multi-turn investigation session without re-sending prior context.

```
$ curl --request POST \
  --url https://api.perplexity.ai/router/v1/responses \
  --header 'Authorization: Bearer <token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "previous_response_id": "<string>",
    "instructions": "Now list the affected CVEs from that
report.",
    "store": true
  }'
```

Run a long-horizon agentic task in the background with a safety identifier, then poll for results, to avoid blocking your pipeline.

```
$ curl --request POST \
  --url https://api.perplexity.ai/router/v1/responses \
  --header 'Authorization: Bearer <token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "instructions": "Enumerate exposed assets and call the scan
tool.",
    "background": true,
    "parallel_tool_calls": true,
    "max_tool_calls": 5,
    "safety_identifier": "pentest-session-42"
  }'
```

— Endpoint, every field, and runnable curl examples given product docs

02 **MCP connector support in Agent API** NEW 83

Agent API requests now support `type: "connector"` using a generated connector ID, letting applications call registered MCP servers without storing or transmitting credentials per-request. Custom MCP servers can be registered via the Project connectors page, supporting API-key or no-auth with Streamable HTTP or SSE transport, available to all Projects. The remote Perplexity MCP Server at `https://api.perplexity.ai/mcp` now supports OAuth, enabling sign-in with a Perplexity account instead of an API key in clients such as claude.ai, Claude Code, Cursor, and VS Code.

Connect an OAuth-capable MCP client to the Perplexity MCP server without managing an API key — authenticate once and choose the billing org during sign-in.

📍 In your MCP client (e.g. Cursor or VS Code), add a new remote MCP server with the URL `https://api.perplexity.ai/mcp`, select OAuth as the auth method, and sign in with your Perplexity account when prompted.

— Names endpoint, transports, and auth flow with setup steps snapshot-20260918

THINNER COVERAGE BELOW

03 **Gemini 3.8 Flash model in Agent API** NEW 55

Adds `google/gemini-3.8-flash` to the Agent API model roster, priced at $0.75/M uncached-input tokens, $0.075/M cached-input tokens, and $3.75/M output and reasoning tokens, through December 31, 2026.

— Pricing and model name given but no usage example snapshot-20260918

snapshot-20260918 NOTES ↗ LATEST STABLE

Perplexity Agent API gains custom MCP connectors, OAuth for remote MCP, and Gemini 3.8 Flash model support.

└──▷ **HOW TO FIND IT**

Connect an OAuth-capable MCP client to the Perplexity MCP server without managing an API key — authenticate once and choose the billing org during sign-in.

📍 In your MCP client (e.g. Cursor or VS Code), add a new remote MCP server with the URL `https://api.perplexity.ai/mcp`, select OAuth as the auth method, and sign in with your Perplexity account when prompted.

Adds `type: "connector"` support in Agent API requests using a generated connector ID, letting applications call registered MCP servers without storing or transmitting credentials per-request.

Adds custom MCP server registration via the Project connectors page, supporting API-key or no-auth with Streamable HTTP or SSE transport, available to all Projects.

Adds OAuth support to the remote Perplexity MCP Server at `https://api.perplexity.ai/mcp`, enabling sign-in with a Perplexity account instead of an API key in clients such as claude.ai, Claude Code, Cursor, and VS Code.

Adds `google/gemini-3.8-flash` to the Agent API model roster, priced at $0.75/M uncached-input, $0.075/M cached-input, and $3.75/M output and reasoning tokens through December 31, 2026.

WAS THIS USEFUL? 👍 👎

Microsoft FLAML

Feedback ⓘ v2.7.0 NOTES ↗ RANK

FLAML automates model selection and hyperparameter tuning under computational resource constraints.

> FLAML v2.7.0 adds per-fold class-imbalance handling to AutoML fitting and extends CI coverage to Windows ARM64.

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

01 **resampler kwarg for per-fold class imbalance** NEW 60

Adds an optional `resampler=` keyword argument to `AutoML.fit` that handles class imbalance on a per-fold basis during cross-validation.

— Names exact kwarg and method but no usage example or mechanism detail. v2.7.0

v2.7.0 NOTES ↗ LATEST STABLE

FLAML v2.7.0 adds a `resampler=` kwarg to `AutoML.fit` for per-fold class-imbalance handling and Windows ARM64 CI support.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.7.0 https://github.com/microsoft/FLAML.git
# already have the repo? check out this version:
$ git checkout v2.7.0
```

Adds optional `resampler=` keyword argument to `AutoML.fit` for per-fold class-imbalance handling during cross-validation.

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.15.0 NOTES ↗ RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> OpenAI Python SDK v3.15.0 adds prompt-cache prewarming and agent session model settings to the API, alongside mentions of managed Responses WebSocket sessions, compaction progress events, and new audio-mini model choices.

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

01 **Prompt-cache prewarming support** NEW 25

Adds prompt-cache prewarming support to the API.

— Bare feature name, no mechanism or usage detail given v3.15.0

02 **Agent session model settings** NEW 25

Adds agent session model settings to the API.

— Bare feature name, no configuration detail given v3.15.0

03 **Managed Responses WebSocket sessions** NEW 20

Adds managed Responses WebSocket sessions, per the release summary.

— Only named in summary, no further detail v3.15.0

04 **Compaction progress events** NEW 15

Adds compaction progress events, per the release summary.

— Only named in summary, no further detail v3.15.0

05 **New audio-mini model choices** NEW 15

Adds new audio-mini model choices, per the release summary.

— Only named in summary, no model names given v3.15.0

v3.15.0 NOTES ↗ LATEST STABLE

OpenAI Python SDK v3.15.0 adds managed Responses WebSocket sessions, prompt-cache prewarming, compaction progress events, and new audio-mini model choices.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.15.0 https://github.com/openai/openai-python.git
# already have the repo? check out this version:
$ git checkout v3.15.0
```

Adds prompt-cache prewarming support to the API.

Adds agent session model settings to the API.

WAS THIS USEFUL?  

OpenAI TypeScript SDK

Feedback ⓘ v7.18.0 NOTES ↗

RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> OpenAI's TypeScript SDK v7.18.0 adds managed WebSocket sessions for the Responses API alongside prompt-cache prewarming, per-session model settings for agent workflows, and new audio-mini model options.

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

01 **Managed Responses WebSocket sessions** NEW 43

Adds managed Responses WebSocket sessions, enabling persistent, server-managed WebSocket connections to the Responses API.

— Names the surface but gives no API or config detail. v7.18.0

02 **Audio-mini model choices** NEW 40

Adds `audio-mini` model choices to the API, expanding the set of selectable models for audio use cases.

— Names the model but no usage example or parameter shown. v7.18.0

03 **Prompt-cache prewarming support** NEW 38

Adds prompt-cache prewarming support, allowing clients to prime the prompt cache before inference requests.

— Describes purpose but no method, flag or endpoint given. v7.18.0

04 **Agent session model settings** NEW 27

Adds agent session model settings, exposing per-session model configuration for agent workflows.

— Bare description with no config keys or usage shown. v7.18.0

v7.18.0 NOTES ↗ LATEST STABLE

OpenAI Node SDK v7.18.0 adds managed Responses WebSocket sessions, prompt-cache prewarming, agent session model settings, and audio-mini model choices.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v7.18.0 https://github.com/openai/openai-node.git
# already have the repo? check out this version:
$ git checkout v7.18.0
```

Adds managed Responses WebSocket sessions, enabling persistent, server-managed WebSocket connections to the Responses API.

Adds prompt-cache prewarming support, allowing clients to prime the prompt cache before inference requests.

Adds agent session model settings, exposing per-session model configuration for agent workflows.

Adds `audio-mini` model choices to the API, expanding the set of selectable models for audio use cases.

WAS THIS USEFUL? 👍 👎

Google JAX

Feedback ⓘ jax-v0.11.2 NOTES ↗ RANK

Composable transformations of Python+NumPy programs: differentiate, vectorize, JIT to GPU/TPU, and more

> JAX 0.11.2 adds mutual-TLS authentication for multi-host coordination, three new numerical primitives, oneMKL LAPACK GPU solvers for Intel hardware, and several breaking changes to sharding, jit inlining, and jnp.tri defaults.

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

01 Mutual TLS for distributed coordination service NEW

95

`jax.distributed.initialize` can now secure its coordination service with mutual TLS via new `mtls_cert_file`, `mtls_key_file`, `mtls_ca_file`, `mtls_peer_uri_prefix`, and `verify_secure_credentials` arguments, or the equivalent `JAX_MTLS_CERT_FILE`, `JAX_MTLS_KEY_FILE`, `JAX_MTLS_CA_FILE`, `JAX_MTLS_PEER_URI_PREFIX`, and `JAX_DISTRIBUTED_VERIFY_SECURE_CREDENTIALS` environment variables, ensuring only authenticated workers can join a multi-host cluster.

Secure a multi-host JAX job's coordination service with mutual TLS so only authenticated workers can join the cluster.

python

```python
import jax

jax.distributed.initialize(
    coordinator_address='coordinator-host:1234',
    num_processes=8,
    process_id=0,
    mtls_cert_file='/etc/certs/client.crt',
    mtls_key_file='/etc/certs/client.key',
    mtls_ca_file='/etc/certs/ca.crt',
    mtls_peer_uri_prefix='spiffe://my-org/jax-worker',
    verify_secure_credentials=True,
)
```

— Names every flag and env var with a runnable example. jax-v0.11.2

02 jax.lax.one_minus_square primitive NEW

80

Adds `jax.lax.one_minus_square` to accurately compute `1 - x^2` near ±1, with accurate derivatives near 0, addressing precision loss in the naive expression.

Compute 1 - x^2 near ±1 with numerically accurate gradients — useful in custom activation functions or physics simulations near boundary values.

```python
import jax
import jax.lax as lax

# accurate result and gradient near x=1.0 where naive (1 - x**2)
loses precision
f = jax.grad(lambda x: lax.one_minus_square(x))
print(f(0.9999999))  # accurate derivative near +1
```

— Explains the numerical fix and includes a runnable gradient example. `jax-v0.11.2`

03 **`jax.numpy.minmax` function** NEW 65

Adds `jax.numpy.minmax` (and `jnp.minmax`) to return both the minimum and maximum of an array in a single call, matching NumPy 2.3+ semantics.

Retrieve both the minimum and maximum of an array in a single pass — useful for normalization or clipping where both bounds are needed.

```python
import jax.numpy as jnp

data = jnp.array([3.0, -1.0, 4.0, 1.0, 5.0, 9.0, 2.0, 6.0])
lo, hi = jnp.minmax(data)
normalized = (data - lo) / (hi - lo)
```

— Named function with runnable example, thin on internals. `jax-v0.11.2`

04 **oneMKL LAPACK solvers for Intel GPUs** NEW 65

Adds oneMKL LAPACK solver kernels for Intel oneAPI GPUs: QR decomposition (`geqrf`, `orgqr` / `ungqr`, `ormqr` / `unmqr`), LU decomposition (`getrf`), symmetric/Hermitian eigenvalue decomposition (`syevd` / `heevd`), SVD (`gesvd`), and hybrid solvers (`geqp3`, `eig`).

— Enumerates every named kernel but gives no usage example. `jax-v0.11.2`

05 **`jax.lax.log2` primitive** NEW 60

Adds `jax.lax.log2` and its `jax.lax.log2_p` primitive, making `log2` a first-class primitive so that `jax.numpy.log2` now lowers via `jax.lax.log2`.

— Names the primitive and lowering path, no usage example. `jax-v0.11.2`

06 **`jax.jit inline=True` default behavior change** BREAKING

60

`inline=True` in `jax.jit` now corresponds to `jax.Inline.JAX_LATE` instead of `jax.Inline.JAX_EARLY`, changing inlining behavior for code that relied on the previous default.

— States exact before/after enum values, no migration example. jax-v0.11.2

THINNER COVERAGE BELOW

07 **Broader input types for remat and generalized_normal** IMPROVED

55

`jax.remat`'s `prevent_cse` argument now accepts `bool | Sequence[bool]`, matching `jax.checkpoint`, and `jax.random.generalized_normal`'s `p` parameter type is widened from `float` to `RealArray`, allowing array-valued shape parameters.

— Names exact type signatures but no runnable example. jax-v0.11.2

08 **Removal of jax.experimental.shard_alike** BREAKING

55

Removed deprecated `jax.experimental.shard_alike`; use explicit sharding mode instead.

— Names the removed API and its replacement approach. jax-v0.11.2

09 **jax.export.symbolic_dim_bounds** NEW

50

Adds `jax.export.symbolic_dim_bounds` for querying conservative bounds on symbolic dimension expressions.

— Names the function but gives no usage detail. jax-v0.11.2

10 **out_sharding parameter for jax.numpy.histogram** IMPROVED

50

Adds `out_sharding` parameter to `jax.numpy.histogram` for controlling output sharding.

— Named flag but no example of use. jax-v0.11.2

11 **jax.numpy.tri default dtype change** BREAKING

50

`jax.numpy.tri` now returns an array with the default float dtype when `dtype` is not specified, instead of always returning `float32`.

— States exact before/after dtype behavior, no example. jax-v0.11.2

12 **Cluster auto-detection for Open MPI 5 and GKE TPU clusters** IMPROVED

45

`jax.distributed.initialize` adds cluster detection support for Open MPI 5, and adds support for reading `TPU_PROCESS_ADDRESSES_PATH` in GKE TPU clusters.

— Names the env var and MPI version but no example. jax-v0.11.2

13 **C++ FFI extension headers in jaxlib wheels** NEW 45

Ships C++ FFI extension headers (`collectives.h` , `record.h`) in `jaxlib` wheels to support out-of-tree plugins.

— Names the header files but no usage example given. jax-v0.11.2

14 **frozendict pytree support for Python 3.15** NEW 40

Adds `frozendict` support to JAX pytrees for Python 3.15 (PEP 814).

— Bare mention of new pytree type support, no example. jax-v0.11.2

15 **exclude_argnames in program_order** IMPROVED 35

Adds `exclude_argnames` argument to `jax.experimental.program_order` .

— Bare name of a new argument, no context given. jax-v0.11.2

└──▷ **BREAKING ON UPGRADE**

! Removed deprecated `jax.experimental.shard_alike` ; use explicit sharding mode instead.

! `inline=True` in `jax.jit` now corresponds to `jax.Inline.JAX_LATE` instead of `jax.Inline.JAX_EARLY` , changing inlining behavior for code that relied on the previous default.

! `jax.numpy.tri` now returns an array with the default float dtype when `dtype` is not specified, instead of always returning `float32` .

jax-v0.11.2 NOTES ↗ LATEST STABLE

JAX 0.11.2 adds mTLS for distributed coordination, new primitives, and oneMKL LAPACK solvers for Intel GPUs.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch jax-v0.11.2 https://github.com/jax-ml/jax.git
# already have the repo? check out this version:
$ git checkout jax-v0.11.2
```

└──▷ **USE IT**

Secure a multi-host JAX job's coordination service with mutual TLS so only authenticated workers can join the cluster.

```python
import jax

jax.distributed.initialize(
    coordinator_address='coordinator-host:1234',
    num_processes=8,
    process_id=0,
    mtls_cert_file='/etc/certs/client.crt',
    mtls_key_file='/etc/certs/client.key',
    mtls_ca_file='/etc/certs/ca.crt',
    mtls_peer_uri_prefix='spiffe://my-org/jax-worker',
    verify_secure_credentials=True,
)
```

Retrieve both the minimum and maximum of an array in a single pass — useful for normalization or clipping where both bounds are needed.

```python
import jax.numpy as jnp

data = jnp.array([3.0, -1.0, 4.0, 1.0, 5.0, 9.0, 2.0, 6.0])
lo, hi = jnp.minmax(data)
normalized = (data - lo) / (hi - lo)
```

Compute 1 - x^2 near ±1 with numerically accurate gradients — useful in custom activation functions or physics simulations near boundary values.

```python
import jax
import jax.lax as lax

# accurate result and gradient near x=1.0 where naive (1 - x**2)
loses precision
f = jax.grad(lambda x: lax.one_minus_square(x))
print(f(0.9999999))  # accurate derivative near +1
```

Secures `jax.distributed.initialize` coordination service with mutual TLS via new `mtls_cert_file`, `mtls_key_file`, `mtls_ca_file`, `mtls_peer_uri_prefix`, and `verify_secure_credentials` arguments, or the corresponding `JAX_MTLS_CERT_FILE`, `JAX_MTLS_KEY_FILE`, `JAX_MTLS_CA_FILE`, `JAX_MTLS_PEER_URI_PREFIX`, and `JAX_DISTRIBUTED_VERIFY_SECURE_CREDENTIALS` environment variables.

Adds `out_sharding` parameter to `jax.numpy.histogram` for controlling output sharding.

Adds `exclude_argnames` argument to `jax.experimental.program_order`.

Updates `jax.remat`'s `prevent_cse` argument to accept `bool | Sequence[bool]`, matching `jax.checkpoint`.

Adds `jax.numpy.minmax` (and `jnp.minmax`) to return both minimum and maximum of an array in a single call, matching NumPy 2.3+.

+9 more

└─▷ **BREAKING ON UPGRADE**

! Removed deprecated `jax.experimental.shard_alike`; use explicit sharding mode instead.

! `inline=True` in `jax.jit` now corresponds to `jax.Inline.JAX_LATE` instead of `jax.Inline.JAX_EARLY`, changing inlining behavior for code that relied on the previous default.

! `jax.numpy.tri` now returns an array with the default float dtype when `dtype` is not specified, instead of always returning `float32`.

WAS THIS USEFUL? 👍 👎

Eigen Labs Darkbloom

Feedback ⓘ v0.9.5 NOTES ↗ CODE ↗

RANK

Private Inference Network on Idle Macs

> Darkbloom v0.9.5 adds native provider support for Qwen 3.8 Flash-Next and forwards warm-pool controller telemetry to Datadog.

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

01 **Native Qwen 3.8 Flash-Next provider support** NEW 85

Adds native Qwen4 provider integration for `qwen3.8-flash-next`, including native model context limits with a 262144-token fallback, tool and reasoning policies, and paging/prefix support. Accessible via the OpenAI-compatible chat completions API by specifying the model name.

Run inference against the new Qwen 3.8 Flash-Next model via the OpenAI-compatible API.

```
$ curl https://api.darkbloom.dev/v1/chat/completions \
  -H 'Authorization: Bearer sk-db-...' \
  -H 'Content-Type: application/json' \
  -d '{"model": "qwen3.8-flash-next", "messages": [{"role":
"user", "content": "Hello!"}], "stream": true}'
```

— Runnable curl example plus named model and token limit v0.9.5

v0.9.5 NOTES ↗ CODE ↗ LATEST STABLE

Darkbloom v0.9.5 adds native Qwen 3.8 Flash-Next (Qwen4) provider support and warm-pool telemetry forwarding to Datadog.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.9.5 https://github.com/Layr-Labs/d-
inference.git
# already have the repo? check out this version:
$ git checkout v0.9.5
```

└──▷ **TRY IT**

Run inference against the new Qwen 3.8 Flash-Next model via the OpenAI-compatible API.

```
$ curl https://api.darkbloom.dev/v1/chat/completions \
  -H 'Authorization: Bearer sk-db-...' \
  -H 'Content-Type: application/json' \
  -d '{"model": "qwen3.8-flash-next", "messages": [{"role": "user",
"content": "Hello!"}], "stream": true}'
```

Adds native Qwen4 provider integration for `qwen3.8-flash-next`, with native model context limits (262144-token fallback), tool and reasoning policies, and paging/prefix support.

Forwards warm-pool controller snapshots (per-model target sizing, demand, capacity blockers, load actions) to Datadog as `warm_pool_tick` events via a new `StartWarmPoolTelemetryLoop`, polling every 15 seconds.

WAS THIS USEFUL?  

Unsloth AI Unsloth

Feedback ⓘ v0.1.810-beta NOTES ↗

RANK

Unsloth runs and fine-tunes language and diffusion models locally through a Python library and interface.

> Unsloth v0.1.810-beta adds AMD ROCm Docker support and RDNA1/RDNA2 GPUs, multi-user accounts with per-account isolation, INT8/FP8 diffusion inference for up to 2x faster image generation, native Windows ARM64 CUDA support, and a string of smaller API, GRPO and interface additions.

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

01 Multi-user accounts with per-account isolation NEW

65

Adds multi-user accounts with per-account isolation, created via `Settings > Accounts` with one-time setup codes and individual passwords.

— Gives exact UI path and mechanism for setup v0.1.810-beta

02 Video input on chat completions API endpoint NEW

60

Adds video input support on the `/v1/chat/completions` API endpoint.

— Exact endpoint named, callable directly v0.1.810-beta

THINNER COVERAGE BELOW

03 AMD GPU support via ROCm Docker and RDNA1/RDNA2 NEW

45

Adds an AMD ROCm Docker image alongside the existing CUDA image for supported Linux hosts, and adds RDNA1 and RDNA2 AMD GPU support.

— Names ROCm/RDNA1/RDNA2 but no setup steps given v0.1.810-beta

04 Qwen3.5 fast_inference and GRPO workflow compatibility NEW

45

Adds Qwen3.5 `fast_inference` support and compatibility with the latest TRL and vLLM in GRPO workflows.

— Names flag and libraries but no example usage v0.1.810-beta

05 **Data Recipes dataset downloads** NEW 45

Adds Data Recipes dataset downloads for building fine-tuning datasets from PDFs, CSVs, and DOCX files.

— Names input formats but no command shown v0.1.810-beta

06 **OS-level tool sandboxing (preview)** NEW 45

Adds preview OS-level tool sandboxing on supported Linux and macOS hosts, with approval requests for file access outside the sandbox.

— Explains approval mechanism, marked preview, no config key v0.1.810-beta

07 **INT8/FP8 image diffusion inference** NEW 40

Adds INT8/FP8 image diffusion inference support, delivering up to 2x faster image generation.

— Concrete speedup number but no usage steps v0.1.810-beta

08 **Windows ARM64 CUDA support** NEW 35

Adds native Windows ARM64 CUDA support for both training and inference.

— States platform scope, no configuration details v0.1.810-beta

09 **MLX video input and MoE/decode optimizations** IMPROVED 35

MLX gains video input support along with optional MoE and decode optimizations.

— Optimizations mentioned but not described mechanistically v0.1.810-beta

10 **Ollama model access through API** NEW 30

Adds access to installed Ollama models through the API.

— Bare statement, no endpoint or config named v0.1.810-beta

11 **GGUF reasoning budgets and interface customization** NEW 30

Adds GGUF reasoning budgets, configurable chat width, and desktop interface scaling.

— Three thin UI/config additions with no settings path v0.1.810-beta

v0.1.810-beta NOTES ↗ PRE-RELEASE

Unsloth v0.1.810-beta adds AMD ROCm Docker, multi-user isolation, INT8/FP8 diffusion, ARM64 Windows CUDA, and GRPO/vLLM improvements.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.810-beta
https://github.com/unslothai/unsloth.git
# already have the repo? check out this version:
$ git checkout v0.1.810-beta
```

Adds an AMD ROCm Docker image alongside the existing CUDA image for supported Linux hosts.

Adds multi-user accounts with per-account isolation, created via `Settings > Accounts` with one-time setup codes and individual passwords.

Adds INT8/FP8 image diffusion inference support, delivering up to 2x faster image generation.

Adds native Windows ARM64 CUDA support for both training and inference.

Adds RDNA1 and RDNA2 AMD GPU support.

+7 more

WAS THIS USEFUL? 👍 👎

Fireworks AI

Feedback ⓘ

New deployment creation flags:deploymentShape: "default"andacceptShapelessRisk

NOTES ↗ RANK

Fireworks AI provides hosted inference, model fine-tuning, and deployment APIs for open-weight models.

> Fireworks AI introduced a deployment shape validation system spanning firectl, the Create Deployment API, and the Python SDK, adding an automatic shape-selection flag, a subcommand to list compatible shapes, and warning that shapeless deployments will soon require an explicit opt-out flag.

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

01 Shapeless deployment opt-out ahead of enforcement

BREAKING

85

Adds `acceptShapelessRisk: true` to the Create Deployment API, `--accept-shapeless-risk` in `firectl`, and `accept_shapeless_risk=True` in the Python SDK as an explicit opt-out to keep creating deployments without a shape, preserving current shapeless behavior. Shapeless creation is the default today, but Fireworks warns that deployments omitting both a shape and this opt-out flag will break once enforcement lands, requiring the explicit flag going forward.

Preserve shapeless deployment behaviour in automation scripts ahead of the upcoming enforcement deadline.

```
$ firectl create deployment --accept-shapeless-risk <model-id>
```

Explicitly opt out of shape validation for an advanced workload not covered by any existing shape via the Create Deployment API.

```
$ curl -X POST https://api.fireworks.ai/v1/deployments \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "<model-id>", "acceptShapelessRisk": true}'
```

— Clear before/after and enforcement deadline with exact flags per interface

New deployment creation flags:deploymen…

02 Automatic deployment shape selection NEW

80

Adds `--deployment-shape default` in `firectl`, `deploymentShape: "default"` in the Create Deployment API, and `deployment_shape="default"` in the Python SDK, letting Fireworks automatically select a validated shape for a model deployment to reduce creation failures. Using the firectl flag requires `firectl` v1.8.8 or newer.

Let Fireworks pick a validated shape automatically when creating a deployment via firectl, to avoid the most common cause of failed deployments.

```
$ firectl create deployment --deployment-shape default --model
<model-id>
```

Create a deployment with auto-selected shape validation using the Python SDK inside an automated provisioning script.

```python
client.create_deployment(model="<model-id>",
deployment_shape="default")
```

Create a deployment with automatic shape selection so Fireworks validates configuration and surfaces conflicts instead of silently failing.

```
$ firectl create deployment --deployment-shape default <model-id>
```

— Names flag, API field, SDK param and version requirement with runnable examples

New deployment creation flags:deploymen…

03 **`deployment-shape-version match` subcommand** NEW 65

Adds `firectl deployment-shape-version match --model <model-id>` to list the deployment shape versions an account can use for a given model, including preset types for cost-per-token optimization, so users can check compatibility before deploying at scale or on a tight budget.

Find cost-optimized shapes for a specific model before deploying at scale or on a tight budget.

```
$ firectl deployment-shape-version match --model <model-id>
```

Check which deployment shape versions your account can use before deploying a model, to avoid incompatible shape selection.

```
$ firectl deployment-shape-version match
```

— Names the subcommand and its purpose with a runnable example product docs

└──▷ **BREAKING ON UPGRADE**

! Shapeless deployment creation (currently the default) will soon require passing `acceptShapelessRisk=true` / `--accept-shapeless-risk` / `accept_shapeless_risk=True` explicitly — deployments that omit both a shape and the opt-out flag will break when enforcement lands.

! Shapeless deployment creation (currently the default) will soon require the explicit `acceptShapelessRisk: true` / `--accept-shapeless-risk` / `accept_shapeless_risk=True` opt-in; existing automation that creates deployments without a shape will break when enforcement lands.

New deployment creation flags:deploymentShape: "default"andacceptShapelessRisk

NOTES ↗ LATEST STABLE

Fireworks AI adds `deploymentShape: "default"` and `acceptShapelessRisk` to the Create Deployment API, firectl, and Python SDK.

└──▷ **TRY IT**

Let Fireworks pick a validated shape automatically when creating a deployment via firectl, to avoid the most common cause of failed deployments.

```
$ firectl create deployment --deployment-shape default --model
<model-id>
```

Create a deployment with auto-selected shape validation using the Python SDK inside an automated provisioning script.

```python
client.create_deployment(model="<model-id>",
deployment_shape="default")
```

Explicitly opt out of shape validation for an advanced workload not covered by any existing shape via the Create Deployment API.

```
$ curl -X POST https://api.fireworks.ai/v1/deployments \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "<model-id>", "acceptShapelessRisk": true}'
```

└──▷ **BREAKING ON UPGRADE**

! Shapeless deployment creation (currently the default) will soon require the explicit `acceptShapelessRisk: true` / `--accept-shapeless-risk` / `accept_shapeless_risk=True` opt-in; existing automation that creates deployments without a shape will break when enforcement lands.

WAS THIS USEFUL? 👍 👎

Ollama

Feedback ⓘ v0.34.2 NOTES ↗ CODE ↗ RANK

Local large language model runner that downloads, serves, and chats with models through a CLI and REST API.

> Ollama v0.34.2 adds a first-run onboarding flow for choosing between cloud sign-in and local-only use, plus a new deep link scheme for jumping straight into the desktop app's Apps page.

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

01 **First-run onboarding flow** NEW 80

Running `ollama` with no subcommand now triggers a first-run setup flow offering sign-in or local-use options. Completion state is shared between CLI and desktop on macOS and Windows, and stored in `~/.ollama/config.json` on Linux.

Trigger the first-run onboarding flow on a fresh Linux install to choose between cloud sign-in and local-only use.

```
$ ollama
```

— Names exact config file and cross-platform behaviour, with a runnable trigger. v0.34.2

THINNER COVERAGE BELOW

02 **ollama://apps deep link scheme** NEW 50

Adds an `ollama://apps` URL scheme to open the desktop app's Apps page directly on macOS and Windows.

— Names the exact URL scheme but gives no further mechanism. v0.34.2

v0.34.2 NOTES ↗ CODE ↗ STABLE

Ollama v0.34.2 adds first-run onboarding and an `ollama://apps` deep link for the desktop app.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.34.2 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.34.2
```

└──▷ TRY IT

Trigger the first-run onboarding flow on a fresh Linux install to choose between cloud sign-in and local-only use.

```
$ ollama
```

Adds `ollama://apps` URL scheme to open the desktop app's Apps page directly on macOS and Windows.

Adds first-run setup flow when running `ollama` with no subcommand, offering sign-in or local-use options; completion state is shared between CLI and desktop on macOS and Windows, and stored in `~/.ollama/config.json` on Linux.

WAS THIS USEFUL?  

Modal

Feedback ⓘ announced · Sep 14, 2026 BLOG ↗ RANK

Bring your own code, and run CPU, GPU, and data-intensive compute at scale. The serverless platform for AI and data teams.

> Modal's biggest release in this window adds Sandbox Sidecars and GA VM Sandboxes for more flexible isolated compute, alongside regional proxy selection, environment-level spend budgets, four new Auto Endpoint models, and a faster I/O path that cuts Function call latency.

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

01 Sandbox Sidecars for multi-container workloads NEW

85

New sb._experimental_sidecars.create() API (public alpha) runs additional containers beside a main Sandbox on the same host, connected via a low-latency internal network — useful for custom network proxies, multi-container workloads, and separating agent harnesses from tool execution.

Run a sidecar HTTP proxy or tool server alongside an agent Sandbox — accessible by name over the internal network — to isolate agent harness code from tool execution.

python

```
import modal
app = modal.App.lookup("sidecar-example", create_if_missing=True)
image = modal.Image.debian_slim().build(app)
sb = modal.Sandbox.create("sleep", "600", app=app, image=image,
timeout=300)
sidecar = sb._experimental_sidecars.create(
    "python", "-m", "http.server", "8080",
    name="web",
    image=image,
)
p = sb.exec(
    "python", "-c",
    "import time, urllib.request; time.sleep(1);
print(urllib.request.urlopen('http://web:8080').status)",
)
p.wait()
print(p.stdout.read())  # 200
sb.terminate()
```

— Named API with runnable code example and clear mechanism launch-20260914-fff951a7

02 Four new frontier models on Auto Endpoints NEW

70

Adds Kimi K3, Qwen3.8-2.4T-A95B, GLM-5.3, and GLM-5.3-Flash as supported models on Modal Auto Endpoints, available as Shared Endpoints (token-based pricing) or Dedicated Endpoints (GPU-second billing).

– Names all four models and both pricing modes `launch-20260914-fff951a7`

---

03 **VM Sandboxes reach public beta** IMPROVED 60

VM Sandboxes now run a real Linux kernel, enabling workloads that require Docker-in-Sandbox, eBPF, systemd, cgroups, or custom filesystem mounts.

– Lists concrete supported workloads but no runnable command `launch-20260914-fff951a7`

---

04 **SDK releases with faster filesystem writes and logging** IMPROVED 60

Releases Python SDK v1.5.5 and JavaScript/Go SDKs v0.10.0 with new log APIs, faster Sandbox filesystem writes in JS and Go, and improved autoscaler configuration reporting.

– Names exact SDK versions and constituent improvements `launch-20260914-fff951a7`

THINNER COVERAGE BELOW

05 **Regional proxy selection for static outbound IPs** NEW 55

You can now choose a region when creating a Modal Proxy in workspace settings, giving Functions and Sandboxes static outbound IPs closer to IP-allowlisted resources.

– Names UI location but no exact config key or endpoint `launch-20260914-fff951a7`

---

06 **Environment-level compute-usage budgets** NEW 55

Team and Enterprise workspaces can now set environment-level compute-usage budgets, letting Owners and Managers set a per-Environment spend cap and track current-cycle usage against it.

– Names plan tiers and roles but no config surface `launch-20260914-fff951a7`

---

07 **Restricted Environments with default role control** NEW 55

Restricted Environments now support setting a default Role (including No Access), enabling private Environments within a shared workspace where non-granted Members cannot discover or access the Environment.

– Names the `No Access` role option and behaviour `launch-20260914-fff951a7`

---

08 **Sandbox directory snapshots and filesystem API GA** IMPROVED 55

Sandbox directory snapshots and the new Sandbox filesystem API are now GA, offering snapshot of selected directories (rather than the full container filesystem) and improved reliability and performance for small files.

— Explains GA scope and before/after but no API names launch-20260914-fff951a7

09 **Lower latency Function calls via rebuilt I/O path** IMPROVED

40

Delivers ~80ms lower p50 latency for Function calls via a rebuilt I/O path with a faster, geographically distributed routing layer.

— Gives a concrete latency number but no user action launch-20260914-fff951a7

10 **Reduced pricing multiplier for broad region selection**

IMPROVED

40

Reduces the pricing multiplier for broad region selection from 1.5x to 1.15x.

— Exact before/after numbers but thin on mechanism launch-20260914-fff951a7

11 **Light mode and refreshed dashboard navigation** NEW

35

Adds a light mode appearance option (beta) and refreshed navigation and color system to the Modal dashboard.

— Brief UI description with no specifics beyond beta toggle launch-20260914-fff951a7

announced · Sep 14, 2026 BLOG ↗

Modal adds Sandbox Sidecars, VM Sandboxes GA, regional proxies, environment budgets, and four new frontier models on Auto Endpoints

└──▷ **USE IT**

Run a sidecar HTTP proxy or tool server alongside an agent Sandbox — accessible by name over the internal network — to isolate agent harness code from tool execution.

```python
import modal
app = modal.App.lookup("sidecar-example", create_if_missing=True)
image = modal.Image.debian_slim().build(app)
sb = modal.Sandbox.create("sleep", "600", app=app, image=image,
timeout=300)
sidecar = sb._experimental_sidecars.create(
    "python", "-m", "http.server", "8080",
    name="web",
    image=image,
)
p = sb.exec(
    "python", "-c",
    "import time, urllib.request; time.sleep(1);
print(urllib.request.urlopen('http://web:8080').status)",
)
p.wait()
print(p.stdout.read())  # 200
sb.terminate()
```

Adds sb._experimental_sidecars.create() (Sandbox Sidecars, public alpha) to run additional containers beside a main Sandbox on the same host, connected via a low-latency internal network — useful for custom network proxies, multi-container workloads, and separating agent harnesses from tool execution.

Adds VM Sandboxes (public beta) that run a real Linux kernel, enabling workloads that require Docker-in-Sandbox, eBPF, systemd, cgroups, or custom filesystem mounts.

Adds regional Proxy selection so you can choose a region when creating a Modal Proxy in workspace settings, giving Functions and Sandboxes static outbound IPs closer to IP-allowlisted resources.

Adds environment-level compute-usage budgets for Team and Enterprise workspaces, letting Owners and Managers set a per-Environment spend cap and track current-cycle usage against it.

Adds support for setting a default Role (including `No Access`) on Restricted Environments, enabling private Environments within a shared workspace where non-granted Members cannot discover or access the Environment.

+6 more

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

vMLX

Feedback ⓘ v1.6.62 NOTES ↗ RANK

The vMLX server runs compressed MLX models on Apple Silicon with disk caching, paged memory, continuous batching, and hybrid SSM scheduling.

> vMLX 1.6.62 adds a new Bonsai2 Hadamard and packed ternary compute runtime and a wide set of opt-in fusion kernels for the Qwen and GLM model families.

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

01 **Opt-in AR fusion kernels for Qwen and GLM** NEW 65

For Qwen: adds exact Qwen AR down and HC boundary fusions, a mixed-dtype Qwen GDN prework fusion, a precise AR sigmoid epilogue for Qwen4, and bounded cold-page hints for Qwen PLE, all opt-in, plus enables guarded Qwen AR fusions with explicit opt-outs. For GLM: adds guarded GLM compound mHC normalization for AR decode and guarded GLM router and shared projection dispatch, enables qualified GLM AR compound normalization with an explicit opt-out, and admits hydrated GLM coefficients while exposing compound graph counts.

— Many named kernels and opt-in/opt-out behaviour, but no exact flag or config key given.

v1.6.62

THINNER COVERAGE BELOW

02 **Bonsai2 Hadamard and packed ternary runtime** NEW 40

Adds the Bonsai2 Hadamard and packed ternary runtime with guarded loading.

— Names the runtime but no flag, config key, or command given. v1.6.62

v1.6.62 NOTES ↗ LATEST STABLE

vMLX 1.6.62 adds Bonsai2 Hadamard and packed ternary runtime plus opt-in Qwen and GLM AR fusion kernels.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.6.62 https://github.com/jjang-ai/vmlx.git
# already have the repo? check out this version:
$ git checkout v1.6.62
```

Adds Bonsai2 Hadamard and packed ternary runtime with guarded loading.

> Adds opt-in exact Qwen AR down and HC boundary fusions.
>
> Adds opt-in mixed-dtype Qwen GDN prework fusion.
>
> Adds opt-in precise AR sigmoid epilogue for Qwen4.
>
> Adds bounded opt-in cold-page hints for Qwen PLE.
>
> +5 more

WAS THIS USEFUL? 👍 👎

LocalAI

Feedback ⓘ v4.10.0 NOTES ↗ CODE ↗ RANK

LocalAI runs language models locally through self-hosted, OpenAI-compatible APIs.

> LocalAI v4.10.0 ships a fleet operations dashboard for multi-node clusters, a credentials file for authenticated model downloads, a new `local-ai benchmark` CLI, per-model environment variables, and expanded LoRA and capabilities-API support.

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

01 **Credentials file for authenticated downloads** NEW 95

Adds `--credentials-file` / `LOCALAI_CREDENTIALS_FILE` to load a `credentials.yaml` that authenticates OCI registries, galleries, HTTP downloads, and `github:` URIs via basic auth (`username` / `password` / `password_env` / `password_file`), bearer tokens (`bearer` / `bearer_env` / `bearer_file`), or custom headers (`header.name` / `header.value` / `_env` / `_file`); secrets are read at use time so rotated Kubernetes secret mounts take effect without a restart.

Authenticate a private OCI registry and a self-hosted gallery without scattering tokens in environment variables.

yaml

```yaml
- url_prefix: https://registry.example.com
  basic:
    username: myuser
    password_env: REGISTRY_PASSWORD

- url_prefix: https://gallery.internal/
  bearer:
    bearer_env: GALLERY_TOKEN

- url_prefix: https://artifacts.corp/
  header:
    name: X-Api-Key
    value_env: CORP_API_KEY
```

— Flag, env var and every auth field named with a runnable config example v4.10.0

02 **`local-ai benchmark` CLI command** NEW 95

Adds `local-ai benchmark MODEL [MODEL ...]`, measuring end-to-end latency and throughput against text models, with `--runs` (default 3), `--warmup` (default 1),

`--max-tokens` (default 128), `--timeout` (default 5m), and `--json` output alongside the default table format.

Measure throughput and latency across multiple models before deploying to production, with structured output for CI ingestion.

```
$ local-ai benchmark llama-3.2-1b-instruct:q4_k_m mistral-7b-instruct:q4_k_m --runs 5 --warmup 2 --max-tokens 256 --json
```

— Exact command with all flags and defaults plus example v4.10.0

---

03 **Per-model environment variables** NEW 82

Adds an `env:` map to model config that injects per-model environment variables (e.g. `CUDA_VISIBLE_DEVICES`, `VLLM_CACHE_DIR`) directly into the backend process, without wrapping it in a script.

Pin a specific GPU and cache directory per model so two models on the same host use different devices without wrapper scripts.

```yaml
name: my-vllm-model
backend: vllm
model: /models/llama-3.2-70b.gguf
env:
  CUDA_VISIBLE_DEVICES: '1'
  VLLM_CACHE_DIR: /mnt/fast-nvme/vllm-cache
```

— Config key named with concrete example values v4.10.0

---

04 **Fleet operations dashboard on Nodes page** NEW 80

Adds a fleet operations dashboard to the Nodes page with a ClusterOverview (health bands, VRAM/RAM/CPU/disk capacity gauges), a NodeFleetTable (sortable, groupable, paginated at 50 rows, bulk Drain/Resume/Remove with concurrency limit 8), a Running Models tab showing replica placement across workers, and slide-out NodeInspector/ModelInspector panels. Workers now report `cpu_logical_cores`, `cpu_usage_percent`, and `cpu_load_1` at registration and heartbeat, surfacing CPU telemetry in the dashboard.

— Detailed UI components named, reachable via Nodes page navigation v4.10.0

---

05 **`context_size` in models capabilities API** IMPROVED 78

Exposes `context_size` in `GET /v1/models/capabilities` responses, letting clients budget tokens and truncate prompts without querying model config out-of-band.

— Endpoint and field named but no request/response example v4.10.0

06 **LoRA adapter fusion in vllm-cpp backend** NEW 78

Adds `lora_adapters`, `lora_scales`, `lora_adapter`, and `lora_scale` config field support to the vllm-cpp backend, fusing LoRA deltas into DiT weights at engine load for LTX2.5 and MiniMax-H3.

— Config fields and target models named, no example config shown v4.10.0

07 **System message merging control in templates** NEW 65

Adds `template.system_messages_after_first` to model config, controlling how system-role messages after the leading system block are handled: `'merge'` folds them into the first system message, `'user'` rewrites them to role `user`.

— Config field and both modes named but no usage example v4.10.0

THINNER COVERAGE BELOW

08 **140 new model gallery entries** NEW 45

Adds 140 new model gallery entries, bringing the total to 1,847, including Qwopus 3.6, LFM2.5 DSpark, Qwen3.8 distills, Granite 4.2, and Gemma 4 E4B.

— Named models and count but no usage detail v4.10.0

09 **ROCm build of stablediffusion-ggml backend** NEW 35

Adds a ROCm build of the `stablediffusion-ggml` backend.

— Bare one-line addition with no further detail v4.10.0

v4.10.0 NOTES ↗ CODE ↗ LATEST STABLE

LocalAI v4.10.0 adds a fleet dashboard, credentials file, `local-ai benchmark` CLI, per-model env vars, and `context_size` in the capabilities API.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.10.0 https://github.com/mudler/LocalAI.git
# already have the repo? check out this version:
$ git checkout v4.10.0
```

└──▷ TRY IT

Measure throughput and latency across multiple models before deploying to production, with structured output for CI ingestion.

```
$ local-ai benchmark llama-3.2-1b-instruct:q4_k_m mistral-7b-instruct:q4_k_m --runs 5 --warmup 2 --max-tokens 256 --json
```

Authenticate a private OCI registry and a self-hosted gallery without scattering tokens in environment variables.

credentials.yaml

```
- url_prefix: https://registry.example.com
  basic:
    username: myuser
    password_env: REGISTRY_PASSWORD

- url_prefix: https://gallery.internal/
  bearer:
    bearer_env: GALLERY_TOKEN

- url_prefix: https://artifacts.corp/
  header:
    name: X-Api-Key
    value_env: CORP_API_KEY
```

Pin a specific GPU and cache directory per model so two models on the same host use different devices without wrapper scripts.

yaml

```
name: my-vllm-model
backend: vllm
model: /models/llama-3.2-70b.gguf
env:
  CUDA_VISIBLE_DEVICES: '1'
  VLLM_CACHE_DIR: /mnt/fast-nvme/vllm-cache
```

Adds `--credentials-file` / `LOCALAI_CREDENTIALS_FILE` to load a `credentials.yaml` authenticating OCI registries, galleries, HTTP downloads, and `github:` URIs via basic auth (`username` / `password` / `password_env` / `password_file`), bearer tokens (`bearer` / `bearer_env` / `bearer_file`), or custom headers (`header.name` / `header.value` / `_env` / `_file`); secrets are read at use time so rotated K8s secret mounts take effect without restart.

Adds `local-ai benchmark MODEL [MODEL ...]` CLI command measuring end-to-end latency and throughput against text models, with `--runs` (default 3), `--`

`warmup` (default 1), `--max-tokens` (default 128), `--timeout` (default 5m), and `--json` output alongside the default table format.

Adds `env:` map to model config, injecting per-model environment variables (e.g. `CUDA_VISIBLE_DEVICES`, `VLLM_CACHE_DIR`) into the backend process without wrapping it in a script.

Adds `template.system_messages_after_first` to model config, controlling how system-role messages after the leading system block are handled: `'merge'` folds them into the first system message, `'user'` rewrites them to role `user`.

Exposes `context_size` in `GET /v1/models/capabilities` responses, letting clients budget tokens and truncate prompts without querying model config out-of-band.

+5 more

WAS THIS USEFUL? 👍 👎

LM Studio

Feedback ⓘ

Session References and Introspection (2026-09-17) BLOG ↗ RANK

LM Studio is a desktop application that runs local LLMs and provides model discovery, downloads, chat, and a local API server.

> LM Studio's Bionic agent gained an Introspection capability that lets it search and read its own persisted session transcripts, including cross-session lookups via @ mentions in the composer.

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

01 Introspection tools for Bionic agent transcripts NEW

82

Introduces Introspection — a set of progressively disclosed tools and a built-in skill letting the Bionic agent read and search its own persisted session transcripts to recover details lost during compaction. The design is tiered: a high-level transcript search with truncation runs first, followed by full message retrieval as a separate tool call, with options to filter out tool call results. Cross-session references are supported via @ mention syntax in the composer, letting users point the agent at a specific past session from any project; the agent requires explicit user permission before reading any session other than the current one, to prevent unintended cross-session contamination.

Recover a design decision that was lost to compaction mid-way through a multi-hour task by letting the agent introspect the current session transcript automatically.

> 📍 In the Bionic composer, continue your task as normal. When the agent detects a gap in its knowledge (e.g., a forgotten design constraint), it will automatically invoke the Introspection skill to search the persisted transcript and retrieve the relevant message without any extra user action.

Pull context from a previous project session into the current one by referencing it directly in your prompt.

> 📍 In the Bionic composer, type @ and select the past session you want to reference — e.g., type: '@<past-session> What auth approach did we settle on?' — to give the agent permission to read that session and retrieve the relevant details.

When a long Bionic session has undergone compaction and a design decision is no longer in active context, type an @ mention of the relevant past session in the composer to let the agent search and recover that detail.

📍 In the Bionic composer, type @ and select the past session you want to reference. Bionic will prompt for permission before reading that session's transcript.

— Names mechanism, tiered tool design, syntax and permission flow clearly.

launch-20260917-aaff8401 | Session References and Introspection (2…

Session References and Introspection (2026-09-17) BLOG ↗ LATEST STABLE

LM Studio Bionic gains Introspection tools to search past session transcripts and cross-session @ references in the composer.

└──▷ HOW TO FIND IT

Recover a design decision that was lost to compaction mid-way through a multi-hour task by letting the agent introspect the current session transcript automatically.

📍 In the Bionic composer, continue your task as normal. When the agent detects a gap in its knowledge (e.g., a forgotten design constraint), it will automatically invoke the Introspection skill to search the persisted transcript and retrieve the relevant message without any extra user action.

Pull context from a previous project session into the current one by referencing it directly in your prompt.

📍 In the Bionic composer, type @ and select the past session you want to reference — e.g., type: '@<past-session> What auth approach did we settle on?' — to give the agent permission to read that session and retrieve the relevant details.

Introduces `Introspection` — a set of progressively disclosed tools and a built-in skill that lets the Bionic agent read and search its own persisted session transcripts to recover details lost during compaction.

Adds cross-session reference support via `@` mention syntax in the composer, allowing users to point the agent at a specific past session from any project.

Requires explicit user permission when the agent reads sessions other than the current one, preventing unintended cross-session contamination.

announced · Sep 17, 2026 BLOG ↗

LM Studio: Supports cross-session referencing via `@` mention in the composer, letting the agent read transcripts from other sessions (including across projects) with explicit permission.

└──▷ HOW TO FIND IT

When a long Bionic session has undergone compaction and a design decision is no longer in active context, type an `@` mention of the relevant past session in the composer to let the agent search and recover that detail.

> 📍 In the Bionic composer, type @ and select the past session you want to reference. Bionic will prompt for permission before reading that session's transcript.

Implements tiered, progressively disclosed tool design for introspection: high-level transcript search with truncation first, then full message retrieval as a separate tool call, with options to filter out tool call results.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Milvus

Feedback ⓘ client/v3.0.0 NOTES ↗ RANK

Milvus is a high-performance, cloud-native vector database built for scalable vector ANN search

> Milvus's Go client v3.0.0 is a major, breaking rewrite that moves to a new module path and proto version while adding durable snapshot export/restore, external collections, search-time aggregation, membership filter blobs, TEXT fields, several new index types, and default-on client telemetry.

└──▷ WHAT SHIPPED · 22 FEATURES most completely described first ? what's the number?

01 **Migration to client/v3 module and proto** BREAKING 90

The Go module moves to `github.com/milvus-io/milvus/client/v3` (install via `go get github.com/milvus-io/milvus/client/v3@v3.0.0`), requiring all imports to be updated from `client/v2`. Generated proto messages must migrate from `milvus-proto/go-api/v2` to `milvus-proto/go-api/v3` (not interchangeable with v2), the module now requires Go 1.24.9 or later (1.24.12 recommended), and this client is not guaranteed compatible with Milvus 2.6 servers.

— Exact package paths, install command and version requirements given client/v3.0.0

02 **Snapshot export and restore management** BREAKING 90

Adds `ExportSnapshot` (now returns an `int64` job ID rather than a path string), `GetExportSnapshotState`, `GetRestoreSnapshotState`, and `RestoreExternalSnapshot` for durable asynchronous snapshot export/restore, plus collection snapshot APIs to create, drop, list, describe, pin, and unpin snapshots and to query and list asynchronous restore jobs. Breaking: callers relying on `ExportSnapshot`'s old string return must switch to polling `GetExportSnapshotState`.

— Named methods and explicit before/after of return type client/v3.0.0

03 **Client-side membership filter blobs** NEW 90

Adds NewBloomFilterBlob(members, fpr) and NewRoaringBitmapBlob(members) in new packages `client/membership/sbbf` and `client/membership/roaringfilter`, used to build `membership_match` expression filter blobs passed into search via `WithTemplateParam`.

Filter a large allow-list at search time without hitting the proxy gRPC receive limit by building a Bloom filter blob client-side.

```go
import (
    "github.com/milvus-io/milvus/client/v3/membership/sbbf"
)

// Build a Split-Block Bloom Filter for 1 million member IDs at
1% FPR
blob, _ := sbbf.NewBloomFilterBlob(memberIDs, 0.01)

results, _ := client.Search(ctx,
milvus.NewSearchOption("my_collection", 10, queryVec).
    WithTemplateParam("membership_match", blob).
    WithFilter("membership_match($meta, {membership_match})"))
```

— Named packages, functions and a runnable example `client/v3.0.0`

04 **Query result ordering** NEW 80

Adds `QueryOption.WithOrderByFields` to order query results with expressions such as `price:desc` and `name:asc`; requires an explicit query limit to be set.

Order query results by a numeric field and return the top 20 to power a ranked leaderboard or price-sorted product listing.

```go
results, _ := client.Query(ctx,
milvus.NewQueryOption("products").
    WithFilter("category == 'electronics'").
    WithOutputFields("name", "price").
    WithOrderByFields("price:desc").
    WithLimit(20))
```

— Named flag with runnable code example `client/v3.0.0`

05 **Client telemetry enabled by default** BREAKING 80

Client telemetry (request counts, errors, P99 latency, server heartbeats) is now enabled by default; set `ClientConfig.TelemetryConfig.Enabled` to `false` to disable heartbeat and metrics reporting.

Disable the new default client telemetry when operating in an air-gapped or privacy-sensitive environment.

```go
cfg := milvus.ClientConfig{
    Address: "localhost:19530",
    TelemetryConfig: milvus.TelemetryConfig{
        Enabled: false,
    },
}
client, _ := milvus.NewClient(ctx, cfg)
```

— Default-on breaking change with exact disable code `client/v3.0.0`

06 **External collection support** NEW 70

Adds external collection support via `Schema.WithExternalSource`, `Schema.WithExternalSpec`, and `Field.WithExternalField`, with `RefreshExternalCollection`, `GetRefreshExternalCollectionProgress`, and `ListRefreshExternalCollectionJobs` for tracking refresh jobs.

— Several named methods, no walkthrough given `client/v3.0.0`

07 **Search result aggregation** NEW 70

Adds search aggregation through bucket aggregation builders supporting multi-field keys, nested aggregations, top-hits, ordering, and avg/sum/count/min/max metrics, with results returned via `ResultSet.AggregationBuckets`.

— Describes mechanism and result field but no code sample `client/v3.0.0`

08 **Structured RPC error inspection** NEW 65

Adds structured RPC error inspection via public `RPCError`, `ErrorCode`, `IsRetryableError`, and common error sentinels compatible with `errors.Is`.

— Named types, no usage example `client/v3.0.0`

09 **Function-based search scoring** NEW 60

Adds `SearchOption.WithFunctionScore` and `HybridSearchOption.WithFunctionScore`, carrying scoring functions with `boost_mode` and `function_mode` parameters applied at search time.

— Named params, no example given `client/v3.0.0`

10 **Namespace-scoped operations** NEW 60

Adds `WithNamespace` support to Search, Hybrid Search, Query, iterators, Insert, Upsert, and Delete options for namespace-scoped operations.

— Named flag across many operations, no example `client/v3.0.0`

11 **TEXT field type** NEW 60

Adds `FieldTypeText` (DataType 25), `column.NewColumnText` / `NewNullableColumnText`, and `WithTextColumn` for variable-length TEXT fields that do not require a `max_length`.

— Named type and constructors, no example `client/v3.0.0`

12 **New index type builders** NEW 60

Adds index type builders `NewHNSWSQIndex`, `NewHNSWPQIndex`, `NewHNSWPRQIndex`, `NewAISAQIndex`, `NewNgramIndex`, and `NewFMIndex`, plus `WithExtraIndexParams` for merging raw build parameters.

— Six named builders plus a param helper, little mechanism `client/v3.0.0`

THINNER COVERAGE BELOW

13 **Search using existing entities as input** NEW 55

Adds `NewSearchByIDsOption` and `AnnRequest.WithIDs` to run vector search using existing entities as the query input, supporting both Int64 and VarChar primary keys.

— Named methods but no runnable example `client/v3.0.0`

14 **Dynamic StructArray fields** NEW 55

Adds `Client.AddCollectionStructField` and `NewAddCollectionStructFieldOption` for dynamic StructArray fields with nullable and `max_capacity` metadata.

— Named methods and metadata options, no example `client/v3.0.0`

15 **Weighted RRF reranking in hybrid search** NEW 50

Adds `NewRRFReranker().WithWeights` for per-ANN-request reciprocal-rank coefficient weighting in hybrid search.

— Named method, minimal mechanism detail `client/v3.0.0`

16 **Collection schema alteration** NEW 50

Adds `DropCollectionField`, `AddFunctionField`, and `DropFunctionField`, all routed through `AlterCollectionSchema` for schema alteration.

— Named methods, thin mechanism description `client/v3.0.0`

17 **File resource management** NEW 50

Adds `AddFileResource`, `ListFileResources`, and `RemoveFileResource` APIs for managing remote files registered with Milvus.

— Named CRUD APIs, no further mechanism given `client/v3.0.0`

18 **Two-phase bulk import control** NEW 50

Adds `CommitImport` and `AbortImport` REST wrappers for two-phase commit control of bulk-import jobs.

— Named endpoints, minimal detail on workflow `client/v3.0.0`

19 **RBAC role descriptions and force-drop** NEW 50

Adds RBAC user and role descriptions, role-description updates via `AlterRole`, and force-drop support for roles.

— Named method, thin description of RBAC changes `client/v3.0.0`

20 **Client request ID correlation** NEW 45

Adds `WithClientRequestID` / `NewClientRequestID` for opt-in request ID correlation with server logs and traces.

— Named functions, thin mechanism `client/v3.0.0`

21 **Message streaming for replication** NEW 35

Adds `Client.DumpMessages`, a streaming wrapper for replication and recovery workflows.

— One named method, minimal explanation `client/v3.0.0`

22 **MinHash function support** NEW 30

Adds `FunctionTypeMinHash` support to the Go SDK function API.

— Single named type, no further detail `client/v3.0.0`

└──▷ **BREAKING ON UPGRADE**

! All imports must be updated from `github.com/milvus-io/milvus/client/v2` to `github.com/milvus-io/milvus/client/v3`.

! Applications using generated Milvus proto messages must migrate from `milvus-proto/go-api/v2` to `milvus-proto/go-api/v3`; v2 and v3 protobuf types are not interchangeable.

! `ExportSnapshot` return type changed from a snapshot path string to an `int64` export job ID; callers must switch to `GetExportSnapshotState` to poll job progress.

! Client telemetry is enabled by default; set `ClientConfig.TelemetryConfig.Enabled` to `false` to disable heartbeat and metrics reporting.

! The module requires Go 1.24.9 or later (Go 1.24.12 recommended); older Go versions are unsupported.

! This client is not guaranteed to be compatible with Milvus 2.6 servers.

`client/v3.0.0` NOTES ↗ LATEST STABLE

Milvus Go client v3 adds snapshots, external collections, search aggregation, membership filters, TEXT fields, and new index types

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch client/v3.0.0 https://github.com/milvus-
io/milvus.git
# already have the repo? check out this version:
$ git checkout client/v3.0.0
```

└──▷ **USE IT**

Filter a large allow-list at search time without hitting the proxy gRPC receive limit by building a Bloom filter blob client-side.

```go
import (
    "github.com/milvus-io/milvus/client/v3/membership/sbbf"
)

// Build a Split-Block Bloom Filter for 1 million member IDs at 1%
FPR
blob, _ := sbbf.NewBloomFilterBlob(memberIDs, 0.01)

results, _ := client.Search(ctx,
milvus.NewSearchOption("my_collection", 10, queryVec).
    WithTemplateParam("membership_match", blob).
    WithFilter("membership_match($meta, {membership_match})"))
```

Order query results by a numeric field and return the top 20 to power a ranked leaderboard or price-sorted product listing.

```go
results, _ := client.Query(ctx, milvus.NewQueryOption("products").
    WithFilter("category == 'electronics'").
    WithOutputFields("name", "price").
    WithOrderByFields("price:desc").
    WithLimit(20))
```

Disable the new default client telemetry when operating in an air-gapped or privacy-sensitive environment.

```go
cfg := milvus.ClientConfig{
    Address: "localhost:19530",
    TelemetryConfig: milvus.TelemetryConfig{
        Enabled: false,
    },
}
client, _ := milvus.NewClient(ctx, cfg)
```

Migrates the Go module to `github.com/milvus-io/milvus/client/v3`; install with `go get github.com/milvus-io/milvus/client/v3@v3.0.0`.

Adds `ExportSnapshot` (returns `int64` job ID), `GetExportSnapshotState`, `GetRestoreSnapshotState`, and `RestoreExternalSnapshot` for durable asynchronous snapshot export and restore workflows.

Adds `NewSearchByIDsOption` and `AnnRequest.WithIDs` for using existing entities as vector-search inputs, supporting Int64 and VarChar primary keys.

Adds `QueryOption.WithOrderByFields` for query ordering with expressions such as `price:desc` and `name:asc` (requires an explicit query limit).

Adds `SearchOption.WithFunctionScore` and `HybridSearchOption.WithFunctionScore` carrying scoring functions with `boost_mode` / `function_mode` params at search time.

+18 more

└─▷ **BREAKING ON UPGRADE**

- ! All imports must be updated from `github.com/milvus-io/milvus/client/v2` to `github.com/milvus-io/milvus/client/v3`.
- ! Applications using generated Milvus proto messages must migrate from `milvus-proto/go-api/v2` to `milvus-proto/go-api/v3`; v2 and v3 protobuf types are not interchangeable.
- ! `ExportSnapshot` return type changed from a snapshot path string to an `int64` export job ID; callers must switch to `GetExportSnapshotState` to poll job progress.
- ! Client telemetry is enabled by default; set `ClientConfig.TelemetryConfig.Enabled` to `false` to disable heartbeat and metrics reporting.
- ! The module requires Go 1.24.9 or later (Go 1.24.12 recommended); older Go versions are unsupported.
- ! This client is not guaranteed to be compatible with Milvus 2.6 servers.

WAS THIS USEFUL?  

LanceDB

Feedback ⓘ v0.39.0 NOTES ↗ RANK

LanceDB is an open-source database that stores multimodal data and supports vector search.

> LanceDB's v0.39.0 release and accompanying JS docs added async remote SQL queries, a function management/introspection overhaul (with several breaking API replacements), computed function columns on materialized views, and new JS-side surfaces for function error inspection, OAuth session management, and a Catalog API.

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

01 **Per-row function error inspection** NEW 83

functionErrors() on `Table` lists per-row errors recorded during embedding/function refreshes, returning `FunctionErrors` with `records` and `truncated` fields. Results can be filtered via `FunctionErrorsOptions`' `jobId` field and are paged with a server-default limit of 10,000 rows, capped at 100,000; supporting types `FunctionErrorFragment` and `FunctionErrorRecord` structure the records. Available on LanceDB Cloud and Enterprise only.

After an embedding refresh job runs under a skip policy, inspect which rows failed and why — useful for debugging bad input values that caused the function to error.

```javascript
const result = await table.functionErrors({ jobId: '<job-id>' });
console.log(result.records);
console.log('truncated:', result.truncated);
```

— Names method, types, exact limits and scope product docs

02 **OAuth session lifecycle management** NEW 83

New `OAuthSession` class exposes login(), status(), and logout() for explicit OAuth lifecycle control: login() eagerly runs the configured interactive flow and replaces any cached session, status() reports whether a cached session exists without contacting the identity provider, and logout() removes the local credential cache entry. Configured via `OAuthConfig` fields `issuerUrl`, `clientId`, `scopes`, `flow` (e.g. `OAuthFlowType.DeviceCode`), and `tokenCache.cacheDir`, with supporting types `ClientAuthMethod`, `TokenCacheOptions`, `SessionLogout`, and `SessionStatus`.

Pre-warm the OAuth token cache before opening a LanceDB connection, so subsequent database operations skip the interactive login flow.

```typescript
import { OAuthSession, OAuthFlowType } from '@lancedb/lancedb';

const config = {
  issuerUrl: 'https://issuer.example.com',
  clientId: 'my-app',
  scopes: ['openid', 'offline_access'],
  flow: OAuthFlowType.DeviceCode,
  tokenCache: { cacheDir: '/tmp/my-app/oauth-cache' },
};

const session = new OAuthSession(config);
const status = await session.login();
console.log('Refreshable:', status.refreshable);
```

Check whether a valid cached session already exists before attempting a login, avoiding an unnecessary interactive prompt in a headless environment.

```typescript
const session = new OAuthSession(config);
const status = await session.status();
if (!status.refreshable) {
  await session.login();
}
```

Remove locally cached credentials at the end of a session without revoking the provider token — useful for shared machines where local cleanup is needed.

```typescript
const session = new OAuthSession(config);
const result = await session.logout();
console.log('Credential removed:', result.removed);
```

— Full config fields, methods, and runnable examples product docs

03 **Materialized view creation and cleanup APIs** NEW 75

Adds `options.withNoData` to materialized-view creation to create only the view definition and an empty backing table without populating it. Adds dropMaterializedView() to synchronously drop a named view (it may become unavailable before physical cleanup finishes) and dropMaterializedViewAsync() to start dropping a view and return its cleanup job without waiting.

Drop a materialized view immediately when you no longer need it and want a simple fire-and-forget call.

```javascript
await connection.dropMaterializedView('my_view');
```

Start dropping a large materialized view in the background and monitor or await the cleanup job separately.

```javascript
const job = await
connection.dropMaterializedViewAsync('my_large_view');
await job;
```

— Named methods with behavior and runnable examples `product docs`

04 **Function management and CRUD client APIs** BREAKING 75

Adds `drop_function` to remove registered functions and `list_functions` to enumerate them from the client. Function CRUD routes on the remote client have also been realigned, so clients pinned to the old route paths will break on upgrade.

Audit which functions are registered in your database before cleanup.

```python
functions = await db.list_functions()
for fn in functions:
    print(fn.name, fn.namespace)
```

Remove a stale registered function from the database.

```python
await db.drop_function('my_embedding_fn')
```

— Named client methods, examples, plus breaking route note `v0.39.0`

05 **Function columns on materialized views** BREAKING 70

Materialized views can now include function columns: computed columns defined by registered functions. LanceDB automatically recomputes computed column rows whose inputs changed, and function outputs can now be nullable named outputs. The source namespace is now recorded in a materialized view's definition, a change that may require migration for existing materialized views lacking this field.

— Explains mechanism and migration impact, no exact command `v0.39.0`

06 **Job introspection API replacement** BREAKING 70

`get_job` and `job_history` are replaced by `describe_job` and `query_job_events` for introspecting remote job status and events; any code calling the old methods will break on upgrade.

Check the status and events of a long-running remote SQL job using the new API.

```python
job = await db.describe_job(job_id)
events = await db.query_job_events(job_id)
for event in events:
    print(event.timestamp, event.status)
```

— Named old and new APIs with runnable example v0.39.0

THINNER COVERAGE BELOW

07 **Catalog API for programmatic connections** NEW 50

Adds `connectCatalog` function with `Catalog` and `CatalogOptions` types to programmatically connect to a LanceDB catalog, plus `ListDatabasesResponse` for enumerating databases via the catalog.

— Names the API surface but no usage shown product docs

08 **Stable row IDs no longer auto-enabled on blob tables** BREAKING 45

Stable row IDs are no longer enabled automatically on blob table creation; workloads that relied on that implicit behavior will see different behavior after upgrade.

— States before/after behavior, no migration command v0.39.0

09 **Blob support additions** NEW 35

Adds support for nested blob function signatures and adds blob v2 fetch and field helpers for the Node.js client.

— Names surfaces but no usage detail v0.39.0

10 **Multi-column merge insert on remote tables** IMPROVED 30

Merge insert on remote tables now accepts multiple `on` columns.

— Names the change but no example or limits v0.39.0

11 **Asynchronous remote SQL queries** NEW 20

Adds support for running remote SQL queries asynchronously.

— Bare mention, no API name or example v0.39.0

12 **makeJsonField JS utility** NEW 10

Adds a `makeJsonField` utility function to the JS API.

— Bare name, no explanation of purpose or usage product docs

└──▷ **BREAKING ON UPGRADE**

! `get_job` and `job_history` are replaced by `describe_job` and `query_job_events`; any code calling the old methods will break on upgrade.

! Stable row IDs are no longer enabled automatically on blob table creation; workloads that relied on that implicit behavior will see different behavior after upgrade.

! Function CRUD routes on the remote client have been realigned; clients pinned to the old route paths will break.

! The source namespace is now recorded in materialized view definitions; existing materialized views that lack this field may require migration.

v0.39.0 NOTES ↗ LATEST STABLE

LanceDB v0.39.0 adds async remote SQL queries, drop/list function APIs, multi-column merge insert, and function columns on materialized views.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.39.0 https://github.com/lancedb/lancedb.git
# already have the repo? check out this version:
$ git checkout v0.39.0
```

└──▷ **USE IT**

Audit which functions are registered in your database before cleanup.

```python
functions = await db.list_functions()
for fn in functions:
    print(fn.name, fn.namespace)
```

Remove a stale registered function from the database.

```python
await db.drop_function('my_embedding_fn')
```

Check the status and events of a long-running remote SQL job using the new API.

```python
job = await db.describe_job(job_id)
events = await db.query_job_events(job_id)
for event in events:
    print(event.timestamp, event.status)
```

Adds `drop_function` client API to remove registered functions from the database.

Adds `list_functions` client API to enumerate registered functions.

Replaces `get_job` / `job_history` with `describe_job` / `query_job_events` for job introspection.

Adds asynchronous remote SQL query support.

Accepts multiple `on` columns for merge insert on remote tables.

+6 more

└──▷ **BREAKING ON UPGRADE**

- ! `get_job` and `job_history` are replaced by `describe_job` and `query_job_events`; any code calling the old methods will break on upgrade.
- ! Stable row IDs are no longer enabled automatically on blob table creation; workloads that relied on that implicit behavior will see different behavior after upgrade.
- ! Function CRUD routes on the remote client have been realigned; clients pinned to the old route paths will break.
- ! The source namespace is now recorded in materialized view definitions; existing materialized views that lack this field may require migration.

WAS THIS USEFUL? 👍 👎

Weaviate

Feedback ⓘ announced · Sep 17, 2026 BLOG ↗ 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 1.39 introduces 4-bit Rotational Quantization with optional centering to shrink memory footprint and improve recall, alongside SIMD-accelerated encoding and CPU prefetching optimizations that speed up both search and imports.

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

01 **4-bit Rotational Quantization with centering** NEW 95

Adds `bits: 4` to the `rq` vector index quantizer in `vectorIndexConfig`, enabling 4-bit Rotational Quantization (RQ4) that delivers a 45% heap reduction versus 8-bit RQ. Also adds a `centering: true` option to the same `rq` config (exposed as `centering=True` in the Python client via Configure.VectorIndex.Quantizer.rq()), improving recall@10 by +0.1 to +6.1 percentage points on anisotropic embedding datasets.

Create a collection using 4-bit centered RQ to cut heap usage by ~45% while maintaining high recall on anisotropic embeddings.

python

```python
from weaviate.classes.config import Configure, Property, DataType

client.collections.create(
    name="Recipes",
    vector_config=Configure.Vectors.text2vec_openai(
        quantizer=Configure.VectorIndex.Quantizer.rq(
            bits=4,
            centering=True,
        )
    ),
    properties=[
        Property(name="title", data_type=DataType.TEXT),
    ],
)
```

Configure 4-bit centered RQ directly in the vector index config JSON when using the REST API or schema definition files.

```json
{
  "vectorIndexConfig": {
    "rq": {
      "enabled": true,
      "centering": true,
      "bits": 4
    }
  }
}
```

— Names exact config keys, client call, and quantified heap/recall benchmarks.

launch-20260917-923249be

THINNER COVERAGE BELOW

02 **SIMD-accelerated RQ encoding via FWHT** IMPROVED 55

Adds SIMD-accelerated Fast Walsh-Hadamard Transform (FWHT) encoding on amd64/AVX and arm64/NEON, speeding up RQ8 encoding 3.8× and supporting the new RQ4 encoding at 6.36–8.08 µs on Intel Xeon.

— Concrete hardware and benchmark detail but no user-facing surface to act on.

launch-20260917-923249be

03 **CPU prefetching in HNSW graph traversal** IMPROVED 50

Adds efficient CPU prefetching for AMD64 and ARM64 during HNSW graph traversal, contributing a 7–11% query throughput improvement and 12% faster imports.

— Internal optimization with metrics but no configurable surface.

launch-20260917-923249be

announced · Sep 17, 2026 BLOG ↗

Weaviate 1.39 adds 4-bit Rotational Quantization (RQ4) with optional centering, delivering 45% heap reduction and faster vector search.

└──▷ **USE IT**

Create a collection using 4-bit centered RQ to cut heap usage by ~45% while maintaining high recall on anisotropic embeddings.

```python
from weaviate.classes.config import Configure, Property, DataType

client.collections.create(
    name="Recipes",
    vector_config=Configure.Vectors.text2vec_openai(
        quantizer=Configure.VectorIndex.Quantizer.rq(
            bits=4,
            centering=True,
        )
    ),
    properties=[
        Property(name="title", data_type=DataType.TEXT),
    ],
)
```

Configure 4-bit centered RQ directly in the vector index config JSON when using the REST API or schema definition files.

```json
{
  "vectorIndexConfig": {
    "rq": {
      "enabled": true,
      "centering": true,
      "bits": 4
    }
  }
}
```

Adds `bits: 4` setting to the `rq` vector index quantizer in `vectorIndexConfig`, enabling 4-bit Rotational Quantization (RQ4) with a 45% heap reduction versus 8-bit RQ.

Adds `centering: true` option to the `rq` quantizer config (and `centering=True` in the Python client via Configure.VectorIndex.Quantizer.rq()), improving recall@10 by +0.1 to +6.1 percentage points on anisotropic embedding datasets.

Adds SIMD-accelerated Fast Walsh-Hadamard Transform (FWHT) encoding on amd64/AVX and arm64/NEON, speeding up RQ8 encoding 3.8× and RQ4 encoding is new in 1.39 at 6.36–8.08 µs on Intel Xeon.

Adds efficient CPU prefetching for AMD64 and ARM64 in HNSW graph traversal, contributing 7–11% query throughput improvement and 12% faster imports.

WAS THIS USEFUL?  

Hugging Face Sentence Transformers

Feedback ⓘ

v6.1.0 NOTES ↗ RANK

Sentence Transformers trains and runs embedding, retrieval, and reranking models, and is the reference implementation behind most open embedding models.

> Sentence Transformers v6.1.0 expands multimodal embedding support to handle multiple items per modality and adds SPARSEUP to its pretrained sparse model lineup, alongside breaking changes to how multimodal inputs are ordered and parsed.

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

01 **Multimodal input ordering and video-frame URL breaking changes**

55

BREAKING

For existing multimodal dict inputs, message content now follows each input's key order, which can change embeddings or scores versus prior behavior. Video-frame URLs without recognizable image extensions must now use an explicit frame wrapper to still be treated as a single video.

– Describes concrete behavior change but no migration example given. v6.1.0

02 **Multiple items per modality in multimodal input** IMPROVED

48

Multimodal dict inputs can now contain multiple items per modality in a single input, enabling richer mixed-media embeddings. Chat-template argument routing for multimodal inputs was also improved, including dispatch for `load_audio_from_video`.

– Names the config surface but no usage example or code shown. v6.1.0

03 **SPARSEUP added to pretrained sparse models** NEW

33

SPARSEUP is now listed in the pretrained sparse model tables alongside the existing SPLADE-family models.

– Names the model but gives no detail on its behavior or use. v6.1.0

└──▷ **BREAKING ON UPGRADE**

! For existing multimodal dict inputs, message content now follows each input's key order, which can change embeddings or scores.

! Video-frame URLs without recognizable image extensions must now use an explicit frame wrapper to remain treated as one video.

v6.1.0 NOTES ↗ LATEST STABLE

Sentence Transformers v6.1.0 improves multimodal input handling and adds SPARSEUP to pretrained sparse model coverage.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v6.1.0 https://github.com/huggingface/sentence-transformers.git
# already have the repo? check out this version:
$ git checkout v6.1.0
```

Supports multiple items per modality in a single multimodal dict input, enabling richer mixed-media embeddings.

Adds SPARSEUP to the pretrained sparse model tables alongside existing SPLADE-family models.

Improves chat-template argument routing for multimodal inputs, including `load_audio_from_video` dispatch.

└──▷ BREAKING ON UPGRADE

! For existing multimodal dict inputs, message content now follows each input's key order, which can change embeddings or scores.

! Video-frame URLs without recognizable image extensions must now use an explicit frame wrapper to remain treated as one video.

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.129.0 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 releases add Chandra HTML conversion output with picture alt-text and rich table cell support, new CSV and AFP backends with configurable character encoding, ambient credential support for S3, and a refactored chart extraction pipeline with generic engine options.

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

01 **CSV and AFP backend character-encoding options** NEW

70

Docling adds `CsvBackendOptions` for configuring a new CSV backend and `AfpFormatOption` for AFP documents, both exposing a character-encoding field that accepts any Python codec name (e.g. `shift_jis`, `koi8-r`) with a fallback chain of BOM → UTF-8 → cp1252, giving explicit control over decoding for both formats.

– Named config classes and codecs but no runnable example. product docs

THINNER COVERAGE BELOW

02 **Chart extraction refactored with generic engine options**

IMPROVED

30

Docling refactors chart extraction with generic engine options, giving callers finer control over the chart understanding pipeline.

– States a refactor occurred without naming the new options. v2.129.0

03 **Ambient credential support for S3** NEW

25

Docling v2.129.0 adds ambient credential support for S3.

– Bare mention with no mechanism or usage detail. v2.129.0

v2.129.0 NOTES ↗ LATEST STABLE

Docling v2.129.0 adds ambient credential support for S3 and refactors chart extraction with generic engine options.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.129.0 https://github.com/docling-project/docling.git
# already have the repo? check out this version:
$ git checkout v2.129.0
```

Refactors chart extraction with generic engine options, giving callers finer control over the chart understanding pipeline.

WAS THIS USEFUL? 👍 👎

okf-memory OKF Agent Memory

Feedback ⓘ v0.4.1

NOTES ↗ RANK

Git-native persistent memory for AI coding agents. 2 with sub-300µs in-memory BM25 search, embedded MCP server, and progressive disclosure.

> OKF Agent Memory's v0.4.1 release tightens MCP protocol conformance, adds drift detection for orphaned concepts, validates links inside index files, and hardens multi-agent sync against concurrent writes.

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

01 **`--drift` detection for unindexed concepts** NEW 93

The `okf validate --drift` flag detects concepts that exist on disk but are absent from parent index listings, preventing silent divergence in agent deduplication loops during multi-agent sync runs. Example: `./bin/okf validate knowledge --strict --drift`.

Catch concepts that exist on disk but were never added to a parent index — useful before a multi-agent sync to prevent deduplication drift.

```
$ ./bin/okf validate knowledge --strict --drift
```

— Exact runnable command and clear mechanism given. v0.4.1

02 **Strict MCP conformance for tool outputs** IMPROVED 70

MCP tool execution results now return `structuredContent` structured JSON objects alongside human-readable text blocks, letting agent runtimes consume parsed data deterministically. Tool definitions for `okf_search`, `okf_show`, `okf_validate`, `okf_create`, `okf_update`, and `okf_relate` were aligned with strict JSON Schema `outputSchema` object requirements, resolving schema rejections in OpenCode, Pi agent, and Claude Desktop.

— Names all six tools and three runtimes but no user action. v0.4.1

03 **Dangling link detection in index.md** NEW 65

During graph construction (`LoadBundle`), all Markdown links inside directory `index.md` files are now validated, detecting phantom or dangling references to non-existent concepts.

— Names the function and file but no example given. v0.4.1

THINNER COVERAGE BELOW

04 **Concurrency safety for multi-agent sync** IMPROVED 57

Adds automated conflict reconciliation for concurrent dated log entries in `log.md` across multi-agent sync runs, plus optimistic push retry with exponential backoff for concurrent vault blob uploads.

— Two thin reliability fixes grouped, no direct action shown. v0.4.1

v0.4.1 NOTES ↗ LATEST STABLE

OKF Agent Memory v0.4.1 adds strict MCP conformance, `structuredContent` payloads, `--drift` unindexed-concept detection, and atomic crash-safe writes.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.4.1 https://github.com/okf-memory/okf-agent-memory.git
# already have the repo? check out this version:
$ git checkout v0.4.1
```

└──▷ TRY IT

Catch concepts that exist on disk but were never added to a parent index — useful before a multi-agent sync to prevent deduplication drift.

```
$ ./bin/okf validate knowledge --strict --drift
```

Adds `okf validate --drift` detection of unindexed concepts that exist on disk but are absent from parent index listings, preventing silent divergence in agent deduplication loops.

Returns `structuredContent` structured JSON objects alongside human-readable text blocks in MCP tool execution results, enabling agent runtimes to consume parsed data deterministically.

Aligns MCP tool definitions for `okf_search`, `okf_show`, `okf_validate`, `okf_create`, `okf_update`, and `okf_relate` with strict JSON Schema `outputSchema` object requirements, resolving schema rejections in OpenCode, Pi agent, and Claude Desktop.

Validates all Markdown links inside directory `index.md` files during graph construction (`LoadBundle`), detecting phantom or dangling references to non-existent concepts.

Implements automated conflict reconciliation for concurrent dated log entries in `log.md` across multi-agent sync runs.

+1 more

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20260918 NOTES ↗

RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's biggest changes this window are new thread/run inspection views and an annotation-queue API that now accepts both runs and conversation threads, alongside new CLI evaluator tooling, thread evaluator testing, and the deprecation of legacy feedback-formula and dataset-comparison endpoints.

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

01 **Feedback formula endpoints deprecated** DEPRECATED 85

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

— Exact endpoints, removal date, and migration target named. snapshot-20260918

02 **Thread evaluator testing before saving** NEW 75

The `/runs/rules/validate` endpoint now supports thread evaluators — pass `test_thread_id` and `session_id` to test a multi-turn evaluator against a real conversation before saving. Multi-turn evaluators in the UI gain a matching Test action that runs the evaluator against a sample thread before the rule is saved.

— Endpoint and parameters named but no full request example. snapshot-20260918

03 **CLI LLM-as-judge evaluator creation** NEW 75

Adds the `langsmith evaluator create-llm` CLI command to define structured LLM-as-judge evaluator rules from a prompt, schema, and model config file, targeting a project or dataset.

— Names the command and its inputs but no full invocation shown. snapshot-20260918

04 **Legacy dataset comparison helpers removed** BREAKING 75

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

— Names removed surface and exact replacement endpoint. snapshot-20260918

05 **Evaluator upload replace flag** IMPROVED 70

Adds the `evaluator upload --replace` flag to the LangSmith CLI, updating existing code evaluator rules in place and avoiding a delete-before-create window if the replacement upload fails.

— Flag and failure-safety mechanism named, no example command. snapshot-20260918

THINNER COVERAGE BELOW

06 **Live split assignments column in experiment comparison** NEW 55

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

— Names the column and its live-vs-snapshot behavior, UI-only path. snapshot-20260918

07 **PEP 604 union types in code evaluator uploads** IMPROVED 55

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

— Concrete type-annotation example given, narrow scope. snapshot-20260918

08 **Messages, Turns, and Details views for threads** NEW 50

LangSmith adds Messages, Turns, and Details views for inspecting threads and runs, letting users drill into a multi-turn conversation to inspect each exchange and its metadata without leaving the UI.

Drill into a multi-turn conversation to inspect each exchange and its metadata without leaving the LangSmith UI.

> 📍 In LangSmith, open a thread, then select the Messages, Turns, or Details tab to inspect runs at each level of granularity.

— Names three UI views and navigation steps, no deeper mechanism. product docs

09 **Monthly trace limits per project and user** NEW 30

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

— States the limit exists but no config surface or values. snapshot-20260918

└──▷ BREAKING ON UPGRADE

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

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

snapshot-20260918 NOTES ↗ LATEST STABLE

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

└──▷ **TRY IT**

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

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

Adds `langsmith evaluator create-llm` CLI command to define structured LLM-as-judge evaluator rules from a prompt, schema, and model config file, targeting a project or dataset.

Adds `POST /annotation-queues/<id>/items` endpoint to add RUN items to a single annotation queue; server resolves runs via ClickHouse or SmithDB and returns a standards-shaped items envelope.

Adds `POST /annotation-queues/<id>/items` support for `item_type: THREAD` (requires `thread_id` + `session_id`) to add conversation threads to an annotation queue; mixed RUN and THREAD batches are supported.

The `/runs/rules/validate` endpoint now supports thread evaluators — pass `test_thread_id` and `session_id` to test a multi-turn evaluator against a real conversation before saving.

Adds `evaluator upload --replace` flag to the LangSmith CLI, updating existing code evaluator rules in place and avoiding a delete-before-create window if the replacement upload fails.

+6 more

└──▷ **BREAKING ON UPGRADE**

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

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

WAS THIS USEFUL?  

Arize Phoenix

Feedback ⓘ arize-phoenix-v20.14.0 NOTES ↗

RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix added Harbor benchmark support for evaluating its MCP and CLI tools.

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

01 **Harbor benchmark for MCP and CLI tools** NEW 25

Adds Harbor benchmark support for evaluating Phoenix MCP and CLI tools.

— Names the benchmark but no mechanism or usage detail arize-phoenix-v20.14.0

arize-phoenix-v20.14.0 NOTES ↗ LATEST STABLE

Arize Phoenix v20.14.0 adds a Harbor benchmark for Phoenix MCP and CLI tools evaluation.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch arize-phoenix-v20.14.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout arize-phoenix-v20.14.0
```

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.38.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse v4.38.0 adds metadata presence filters, conversation transcript generation, two-level MCP authorization, and an AI Gateway relay for OpenAI's Responses API, alongside trace viewer UI upgrades and a breaking change to legacy ingestion reads for new organizations.

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

01 **Legacy ingestion GET API rejected for new orgs** BREAKING 65

New organizations now receive HTTP rejections on legacy (non-v2, non-OTel) `GET` ingestion API reads — clients still using the legacy read path will fail on upgrade for newly created organizations.

— Clear before/after and named API surface, but no migration command shown. v4.38.0

02 **OpenAI Responses API relay in AI Gateway** NEW 65

Relays OpenAI Responses API compact events and `GET /models` through the AI Gateway.

— Names endpoint and event type but not full request flow. v4.38.0

THINNER COVERAGE BELOW

03 **Metadata key presence filters** NEW 45

Adds metadata key presence operators to filters, so traces and observations can be queried by whether a metadata key exists, without needing to match on a value.

— Describes mechanism but not the exact operator syntax. v4.38.0

04 **Trace viewer UI improvements** IMPROVED 45

The trace detail view now shows a persistent trace summary strip with tags that stays visible while navigating, and the graph is promoted into a first-class view alongside the existing tree and timeline tabs.

— UI locations named but no config or API surface. v4.38.0

05 **AI Gateway tracing context isolation** IMPROVED 40

Isolates caller tracing context for AI Gateway generations, preventing upstream trace context from leaking into gateway-originated spans.

— Explains the fix mechanism but no reproduction or config detail. v4.38.0

06 **Two-level MCP authorization** NEW 35

Supports two-level MCP authorization across all auth modes, extending the existing auth seam to MCP callers.

— States scope but not the auth modes or configuration steps. v4.38.0

07 **Conversation transcript generation** NEW 35

Generates conversation transcripts from observations, enabling structured replay and review of LLM call sequences.

— Describes purpose but not format or access point for transcripts. v4.38.0

08 **Coding agent request correlation in AI Gateway** NEW 30

Correlates coding agent requests in the AI Gateway for end-to-end visibility into agentic coding workflows.

— Names the feature but not how correlation is implemented or surfaced. v4.38.0

└──▷ BREAKING ON UPGRADE

! New organizations now receive HTTP rejections on legacy (non-v2, non-OTel) GET ingestion API reads — clients still using the legacy read path will fail on upgrade for newly created organizations.

v4.38.0 NOTES ↗ LATEST STABLE

Langfuse v4.38.0 adds metadata key presence filters, transcript generation, MCP two-level auth, and AI Gateway Responses relay.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.38.0
https://github.com/langfuse/langfuse.git
# already have the repo? check out this version:
$ git checkout v4.38.0
```

Adds metadata key presence operators to filters, so you can query traces/observations by whether a metadata key exists — without needing to match on a value.

Generates conversation transcripts from observations, enabling structured replay and review of LLM call sequences.

Supports two-level MCP authorization across all auth modes, extending the auth seam to MCP callers.

Relays OpenAI Responses API compact events and `GET /models` through the AI Gateway.

Shows a persistent trace summary strip with tags in the trace detail view, keeping context visible while navigating.

+3 more

└─▷ **BREAKING ON UPGRADE**

! New organizations now receive HTTP rejections on legacy (non-v2, non-OTel) GET ingestion API reads — clients still using the legacy read path will fail on upgrade for newly created organizations.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20260918 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust's biggest addition this window is Patterns, a scheduled Loop-powered feature that automatically investigates traces for recurring problems, alongside new CLI-driven observability templates and tracing integrations for Google Discovery Engine, Agno, and TypeSafe. It also shipped new project settings for blind reviews and columnstore metrics, Loop collaboration features, collapsible dashboard sections, a faster asynchronous Brainstore log search engine, and removed a deprecated CLI flag.

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

01 **`bt observability template` export/apply commands** NEW

95

Adds `bt observability template` commands in `bt` v0.19.0+ to export a project's configuration as a portable JSON template and apply it to another project, enabling reuse of facets and Loop automations across projects.

— Exact CLI command, format, and version given. snapshot-20260918

02 **Patterns automated trace investigation** NEW

80

Adds Patterns, a scheduled Loop-powered feature that automatically investigates traces for recurring problems and trends, saves findings with supporting traces and suggested fixes, and lets you create a scorer or classifier from any finding; limited to GPT-6 Astra and GPT-5.6 model families; requires data plane v2.13+.

— Detailed mechanism and model/version limits, no exact command. snapshot-20260918

03 **Asynchronous execution engine for Brainstore log search**

IMPROVED

80

Adds an asynchronous execution engine to Brainstore log search that streams results as segments finish scanning, covering full-text, phrase, and multi-clause ANY_SPAN() trace searches including negated clauses; requires data plane v2.13.0+.

— Names search function and version, no runnable example. snapshot-20260918

04 **Loop automations** NEW

70

Adds Loop automations — scheduled runs with their own instruction, model, and write permissions that leave a read-only thread and can send results to a Slack channel or webhook; requires data plane v2.12.0+.

— Names outputs and version requirement, no exact config. snapshot-20260918

05 **Collapsible dashboard sections** NEW 70

Adds dashboard sections that can be named, reordered, duplicated, and collapsed, with collapsed state persisted per browser; available on Pro and Enterprise plans; requires data plane v2.13.0+.

— Names plan tiers and version, no exact UI path. snapshot-20260918

06 **Metric columnstore project setting** NEW 70

Adds Metric columnstore project setting, alongside log search optimization, that writes columnar artifacts during compaction to speed up loading and sorting traces on high-volume projects; requires data plane v2.2.1+.

— Named setting and data-plane version requirement given. snapshot-20260918

07 **TypeSafe Jev tracing support** NEW 70

Adds TypeSafe Jev tracing support to inspect context, question criteria, selected answers, and probability distributions for structured decisions, available in Python SDK v0.41.0 and TypeScript SDK v3.34.0.

— Names inspectable fields and both SDK versions. snapshot-20260918

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

Adds Blind human reviews project setting so peer scores, comments, and aggregates remain hidden for the entire review session to preserve reviewer independence.

— Named setting with clear mechanism, no exact UI path. snapshot-20260918

THINNER COVERAGE BELOW

09 **Loop thread sharing** NEW 55

Adds Loop thread sharing so project members can review or continue an investigation with view-only or chat access; requires data plane v2.15.0+.

— Names access levels and version, no UI path. snapshot-20260918

10 **Google Discovery Engine tracing instrumentation** NEW 55

Adds instrumentation for Google Discovery Engine covering answer generation, conversational search, grounding checks, and ranking, available in TypeScript SDK v3.34.0.

— Names capability areas and SDK version, no code example. product docs

11 **Agno agent framework tracing integration** NEW 45

Adds tracing integration for Agno agents, teams, and workflows to debug prompts, evaluate models, and monitor production usage in Braintrust.

— Describes use cases, no SDK version or command. product docs

└─▷ **BREAKING ON UPGRADE**

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set` in bt v0.20.0 (the flag was deprecated and ignored since data plane v2.15.0); see the v0.20.0 migration instructions.

snapshot-20260918 NOTES ↗ LATEST STABLE

Braintrust: Adds `bt observability template` commands in `bt` v0.19.0+ to export a project's configuration as a portable JSON template and apply it to another project, enabling reuse of facets and Loop automations across projects.

Adds Blind human reviews project setting so peer scores, comments, and aggregates remain hidden for the entire review session to preserve reviewer independence.

Adds Metric columnstore project setting (alongside log search optimization) that writes columnar artifacts during compaction to speed up loading and sorting traces on high-volume projects; requires data plane v2.2.1+.

Adds TypeSafe Jev tracing support — inspect context, question criteria, selected answers, and probability distributions for structured decisions — available in Python SDK v0.41.0 and TypeScript SDK v3.34.0.

Adds Patterns, a scheduled Loop-powered feature that automatically investigates traces for recurring problems and trends, saves findings with supporting traces and suggested fixes, and lets you create a scorer or classifier from any finding; limited to GPT-6 Astra and GPT-5.6 model families; requires data plane v2.13+.

Adds Loop thread sharing so project members can review or continue an investigation with view-only or chat access; requires data plane v2.15.0+.

+3 more

└─▷ **BREAKING ON UPGRADE**

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set` in bt v0.20.0 (the flag was deprecated and ignored since data plane v2.15.0); see the v0.20.0 migration instructions.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.69 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik improved experiment result handling this window, adding full result-set export from both the browser UI and the Python SDK, plus the ability to map trace and span fields onto dataset item columns.

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

01 Full experiment result export via browser and SDK NEW

60

Opik now supports exporting the full experiment result set directly from the browser UI, and adds a Python SDK option to export full experiment results to CSV, matching the columns of the comparison page but without its row cap.

— Names CSV format and row-cap removal but no exact SDK call 2.2.69

THINNER COVERAGE BELOW

02 Trace/span field mapping onto dataset columns NEW

35

Adds the ability to map trace and span fields onto dataset item columns in the experiment view.

— Describes capability but no field names or navigation path 2.2.69

2.2.69 NOTES ↗ LATEST STABLE

Opik 2.2.69 adds full experiment result-set export from the browser and trace/span field mapping onto dataset item columns.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch 2.2.69 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.69
```

Enables exporting the full experiment result set directly from the browser UI.

Adds the ability to map trace and span fields onto dataset item columns in the experiment view.

WAS THIS USEFUL?  

agentacct

Feedback ⓘ v0.11.1 NOTES ↗ RANK

The agentacct dashboard records coding-agent work steps, tools, file changes, tests, time, and token costs locally.

> agentacct v0.11.1 adds disk-reclaiming evidence maintenance, in-place CLI self-updates, a reworked sortable Sessions table in the TUI, and a faster caching model for the work-ledger and sessions APIs.

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

01 **Evidence store pruning command** NEW 85

Adds `agentacct evidence prune` to reclaim disk space from accumulated `tool_activity_observed` and `rate_limit_observed` events previously shadowed into the evidence-v2 store; the operation is transactional, batched, with an optional `VACUUM`.

Reclaim disk space after evidence-v2 has accumulated high-cardinality recorder events from prior versions.

```
$ agentacct evidence prune
```

— Names command, event types, and mechanism (transactional, batched, VACUUM). v0.11.1

02 **CLI self-update command** NEW 75

Adds `agentacct self-update` to upgrade the CLI in place from the Diagnostics pane when a newer release is published; it is refused for development/editable checkouts.

Upgrade the CLI to the latest published release without reinstalling manually.

```
$ agentacct self-update
```

— Runnable command with a clear condition and exception noted. v0.11.1

03 **Sortable Sessions table in TUI** IMPROVED 75

Rebuilds `agentacct tui`'s Sessions master list as a sortable DataTable with Task / Outcome / Evidence / Cost / Age columns; press `s` to cycle sort and `ctrl+d` / `ctrl+u` to scroll the detail pane.

— Names columns and exact keybindings for navigation. v0.11.1

04 **Composite change-key caching for API reduces** IMPROVED

70

Switches the API-serving work-ledger and sessions caches from a 30-second wall-clock TTL to a composite change key (events fingerprint plus append-only signature of secondary stores), warming on startup so idle polls skip the multi-second reduce.

— Detailed mechanism but no user-facing command or flag. v0.11.1

THINNER COVERAGE BELOW

05 **One-click recorder recovery in UI** NEW

50

Adds a Start recorder button to the app's health surfaces and menu-bar dropdown when the recorder is unreachable, enabling one-click recovery.

— Describes trigger and effect but only a UI path. v0.11.1

06 **Help command and friendlier bare invocation** IMPROVED

45

Adds `agentacct help` and a friendly command overview for a bare `agentacct` invocation; the bare invocation now exits 0 instead of a terse error.

— Names the command and exit-code change but little else. v0.11.1

v0.11.1 NOTES ↗ LATEST STABLE

agentacct v0.11.1 adds `agentacct evidence prune`, `agentacct self-update`, and a sortable TUI Sessions table.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.11.1
https://github.com/mikehasa/agentacct.git
# already have the repo? check out this version:
$ git checkout v0.11.1
```

└──▷ TRY IT

Reclaim disk space after evidence-v2 has accumulated high-cardinality recorder events from prior versions.

```
$ agentacct evidence prune
```

Upgrade the CLI to the latest published release without reinstalling manually.

```
$ agentacct self-update
```

Adds `agentacct evidence prune` to reclaim disk space from accumulated `tool_activity_observed` and `rate_limit_observed` events previously shadowed into the evidence-v2 store (transactional, batched, with optional `VACUUM`).

Adds `agentacct self-update` to upgrade the CLI in place from the Diagnostics pane when a newer release is published (refused for development/editable checkouts).

Adds `agentacct help` and a friendly command overview for a bare `agentacct` invocation (bare invocation now exits 0 instead of a terse error).

Rebuilds `agentacct tui`'s Sessions master list as a sortable DataTable with Task / Outcome / Evidence / Cost / Age columns, `s` to cycle sort, and `ctrl+d`/`ctrl+u` to scroll the detail pane.

Adds a Start recorder button to the app's health surfaces and menu-bar dropdown when the recorder is unreachable, enabling one-click recovery.

+1 more

WAS THIS USEFUL? 👍 👎

ai-gateway

Feedback ⓘ v1.5.9 NOTES ↗ RANK

AI Gateway routes AI model requests through a self-hosted gateway with observability and security controls.

> AI Gateway v1.5.9 hardens plugin configuration handling with an environment-expansion kill-switch and adds per-instance plugin identification for clearer rejection diagnostics.

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

01 Environment-expansion kill-switch for plugin configs NEW

85

The new gateway.WithoutEnvExpansion() construction option disables `${VAR}` substitution in plugin configs and MCP server headers/env, preventing untrusted config from resolving against the host process environment.

Build a gateway from operator-supplied config without risking `${VAR}` tokens in that config resolving against the host process environment.

```go
gw, err := gateway.New(cfg, gateway.WithoutEnvExpansion())
```

— Names exact API and shows runnable Go usage v1.5.9

02 Per-instance plugin IDs and duplicate validation NEW

62

Plugin config entries now accept an optional `id` field so operators can identify which specific plugin instance fired; `plugin.RejectionError` carries the instance `id`, and `ferrogw validate` rejects duplicate `id` claims across different instances.

— Names config field, error type and CLI check, no example given v1.5.9

v1.5.9 NOTES ↗ LATEST STABLE

ai-gateway v1.5.9 adds observable guardrail events, per-instance plugin IDs, and an env-expansion kill-switch

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.5.9 https://github.com/ferro-labs/ai-gateway.git
# already have the repo? check out this version:
$ git checkout v1.5.9
```

└──▷ **USE IT**

Build a gateway from operator-supplied config without risking `${VAR}` tokens in that config resolving against the host process environment.

```go
gw, err := gateway.New(cfg, gateway.WithoutEnvExpansion())
```

Adds gateway.WithoutEnvExpansion() construction option to disable `${VAR}` substitution in plugin configs and MCP server headers/env, so untrusted config cannot resolve against the host process environment.

Adds optional `id` field to plugin config entries so operators can identify which specific plugin instance fired — `plugin.RejectionError` now carries the instance `id`, and `ferrogw validate` rejects duplicate `id` claims across different instances.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

agent-safe-pipeline

Feedback ⓘ v0.2.0 NOTES ↗ RANK

Agent Safe Pipeline provides a security-focused framework for building AI agent workflows with policy checks and safety controls.

> Agent Safe Pipeline shipped an HTTP proxy gateway that enforces ALLOW/BLOCK/ESCALATE decisions from the Decionis control plane, promoted its trusted-executor library to the `@decionis/agentsafe` npm package with TLS egress sealing and cryptographic evidence chaining, and rounded this out with CLI tooling, a boundary self-test command, an escalation/approval endpoint, and packaging across five deployment targets.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 **`agentsafe test` boundary self-test command** NEW 89

`agentsafe test` replays consequential requests three ways — direct, shadow, and enforcement — against a synthetic target and reports exposure, evidence, and a BOUNDARY HOLDS / EXPOSURE verdict; accepts `ledger=<host>:<port>` to also probe a real system of record, and a `--json` flag to emit the full boundary report as a single JSON line.

Run the full boundary self-test — including probing a live system of record to confirm an agent could reach it without the gateway — and emit machine-readable results for CI.

```
$ agentsafe test ledger=ledger.internal:443 --json
```

— Names command, flags, and verdict states; runnable example given. product docs

02 **Trusted executor library for in-process enforcement** NEW 86

Exposes a `createTrustedExecutor` API for in-process enforcement covering host posture (`HostPosture`), caller principals, egress sealed to configured origins (`EgressPolicy`), attempt journaling via `StartupReconciler` , hard ceilings (`HardLimits`), a halt switch (`HaltSwitch`), and hash-chained evidence with offline-verifiable export via an `agent-safe.evidence-bundle/1` bundle (hash-chained streams, open attempts, posture by check, chain heads, a configuration digest, and per-file digests). A later release promotes this to the `@decionis/agentsafe` npm package as a first-class deployable process, adds a TLS listener and egress sealing that chains cryptographic evidence across calls, hardens secret handling by treating secrets as files rather than environment strings, adds a principals/roles/credential-management layer, teaches the executor domain-specific behaviour through adapters (including a banking

adapter), adds a verifier that cryptographically signs grants and authority attestations into the incident record, and adds agent-zone containment measurement and reporting.

— Names many APIs and subsystems with real before/after detail. v0.2.0

03 **HTTP gateway proxy with enforcement, shadow, and escalation modes**

NEW 85

Adds `agentsafe proxy --upstream <url> --port <port>` to intercept HTTP requests and enforce ALLOW / BLOCK / ESCALATE decisions from the Decionis control plane before forwarding them, adds a `--verbose` flag to surface chained evidence lines during proxy operation, and stamps forwarded (ALLOW) responses with `agentsafe-decision`, `agentsafe-dossier-id`, and `agentsafe-execution` headers, plus `agentsafe-mode: SHADOW` in shadow mode. A later release adds raw HTTP interception at the gateway and multi-gateway fan-out so one listener can front many gateways by host.

Stand up the enforcement gateway in front of an internal service so every unsafe HTTP method is intercepted and governed before reaching the upstream.

```
$ agentsafe proxy --upstream http://localhost:3000 --port 8080 --verbose
```

— Names flags, headers, and fan-out; runnable command example given. v0.2.0

04 **Escalation resume endpoint and Presence-managed approvals**

NEW 80

The `/_agentsafe/v1/escalations/{intent_id}/resume` endpoint resumes a held (ESCALATE) request once a human approves it via Presence, and the `presence.managed: true` configuration key lets Decionis orchestrate that human-approval ceremony rather than the executor coordinating Presence directly.

Enable managed human-approval orchestration so Decionis coordinates the Presence ceremony for held requests rather than the executor handling it directly.

```yaml
presence:
  managed: true
evidence:
  journalDir: /var/lib/agentsafe/journal
```

— Names endpoint and config key with clear mechanism. product docs

05 **Packaging and deployment across five targets** NEW 75

Ships the runtime across five deployment targets: macOS Homebrew (`brew install agentsafe`), a Linux shell installer and `.deb` / `.rpm` packages with a hardened systemd unit, a Docker image (`ghcr.io/decionis/agentsafe:<version>`), a Helm chart (`oci://ghcr.io/decionis/charts/agentsafe` , also used for Kubernetes deployment), and a hosted endpoint at `agentsafe.decionis.com` . A later release also builds the runtime as a single self-contained executable distributed via the npm tarball `@decionis/agentsafe` , and lets a hosted call declare which package, version, and example it originates from.

Install the agentsafe runtime in one step to get the trusted executor binary on your path.

```
$ curl -fsSL https://raw.githubusercontent.com/decionis/agent-safe-pipeline/v0.2.0/packaging/install.sh | sh
```

Verify the integrity and provenance of the released npm tarball before deploying the executor.

```
$ shasum -a 256 -c SHA256SUMS && gh attestation verify decionis-agent-safe-pipeline-0.2.0.tgz --repo decionis/agent-safe-pipeline --bundle agent-safe-pipeline-0.2.0.provenance.sigstore.json --custom-trusted-root trusted_root.jsonl
```

— Lists every install target and command; thin on mechanism. v0.2.0

06

Gateway evidence journaling and failure-policy configuration

NEW

60

The `evidence.journalDir` configuration key persists chained evidence to a durable journal, and the `failurePolicy` configuration key (values: `failOpen` / fail-closed default) controls gateway behavior when the authority is unreachable.

Enable managed human-approval orchestration so Decionis coordinates the Presence ceremony for held requests rather than the executor handling it directly.

```yaml
presence:
  managed: true
evidence:
  journalDir: /var/lib/agentsafe/journal
```

— Names two config keys but omits deeper behavior detail. product docs

THINNER COVERAGE BELOW

07 **CLI setup commands: init, doctor, login** NEW 50

`agentsafe init` writes the gateway's configuration file, `agentsafe doctor` reports anything that would prevent the gateway from governing traffic, and `agentsafe login` connects a Decionis key so the gateway delegates decisions to Decionis, starting in shadow mode.

— Names three commands but describes each in one line. product docs

08 **Whisper-boundary prompt-injection demo** NEW 40

Adds a Whisper-boundary demo validating six literature prompt-injection attacks produce zero unauthorized effects.

— Single demo statistic, no command or mechanism given. v0.2.0

09 **BEAP 1.0 protocol implementation** NEW 35

Implements BEAP 1.0 identifiers and mirrors the published specification into the runtime, staging the prior BEAP v0.1 specification for archival. The pipeline selects the enforcement gate from the environment so a handler cannot bypass it, carries an expected-effect commitment and the observation that answers it through the pipeline, reports the claim lease and supports in-flight credential rotation, and reports what Decionis decided while verifying the signed record it leaves (the Decision Dossier).

— Names protocol terms but describes mechanics only in prose. v0.2.0

10 **commerce-mcp Claude extension with refund tracking** NEW 35

Adds a `commerce-mcp` MIT Claude extension, and carries refundable balance and refund history on `REFUND_REQUEST` events in the commerce-mcp layer.

— Thin: names extension and event but no mechanism. v0.2.0

v0.2.0 NOTES ↗ LATEST STABLE

agent-safe-pipeline v0.2.0 ships the trusted executor as `@decionis/agentsafe`, adds TLS egress sealing, BEAP 1.0 identifiers, HTTP gateway interception, and a Helm chart for Kubernetes deployment.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.2.0 https://github.com/decionis/agent-safe-pipeline.git
# already have the repo? check out this version:
$ git checkout v0.2.0
```

└──▷ TRY IT

Install the agentsafe runtime in one step to get the trusted executor binary on your path.

```
$ curl -fsSL https://raw.githubusercontent.com/decionis/agent-safe-pipeline/v0.2.0/packaging/install.sh | sh
```

Verify the integrity and provenance of the released npm tarball before deploying the executor.

```
$ shasum -a 256 -c SHA256SUMS && gh attestation verify decionis-agent-safe-pipeline-0.2.0.tgz --repo decionis/agent-safe-pipeline --bundle agent-safe-pipeline-0.2.0.provenance.sigstore.json --custom-trusted-root trusted_root.jsonl
```

Promotes the trusted executor to the `@decionis/agentsafe` npm package, making it a first-class deployable process with its own deployment kit.

Adds TLS listener support and egress sealing to the executor, chaining cryptographic evidence across calls.

Adds host posture verification and hardens secret handling by treating secrets as files rather than environment strings.

Gives the executor a principals, roles, and credential management layer.

Teaches the executor domain-specific behaviour through adapters, including a banking adapter.

+16 more

WAS THIS USEFUL? 👍 👎

ai-safe2-framework

Feedback ⓘ

2026-09-17_CLI_0.6.0_Scope_Attribute_Decide · NOTES ↗ · RANK

Ai-Safe2-Framework provides governance, risk, and compliance controls for securing agentic AI systems and non-human identities.

> AI SAFE² CLI 0.6.0 introduces a trio of `safe2 evidence` subcommands for deployment-boundary scoping, cross-revision finding attribution, and release-readiness decisions, alongside exportable evidence-contract schemas and hardened JUnit XML intake.

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

01 **Release-readiness command and status model** NEW 95

New `safe2 evidence readiness` command produces canonical release-readiness JSON and an escaped Markdown card, accepting `--system-identity`, `--assessment-scope`, `--change-attribution`, `--output`, `--card`, and `--strict` options. Results are classified into three statuses — `hold`, `review`, and `ready_for_human_decision` — with `release_authorized: false` enforced on every result and a named human decision owner required.

— Every flag and status value named with exact command

2026-09-17_CLI_0.6.0_Scope_Attribute_De…

02 **Deployment-boundary scoping command** NEW 78

New `safe2 evidence scope` command inventories the declared deployment relationship of repository or built-artifact paths, classifying each as `included`, `excluded`, `partial`, `not applicable`, or `unknown`.

— Names command and all five classification values

2026-09-17_CLI_0.6.0_Scope_Attribute_De…

03 **Finding attribution across baseline and revision** NEW 78

New `safe2 evidence attribute` command compares normalized findings across an explicitly trusted baseline and the current revision, separating them into `inherited`, `introduced`, `changed`, `resolved`, and `unknown` findings.

— Names command and full set of comparison categories

2026-09-17_CLI_0.6.0_Scope_Attribute_De…

04 **Exportable evidence-contract JSON schemas** NEW 68

`safe2 schema export` now exposes the `assessment-scope-source-v1`, `change-attribution-source-v1`, and `release-readiness-source-v1` JSON

schemas to anchor evidence contracts.

— Names command and all three schema identifiers, thin on mechanism

2026-09-17_CLI_0.6.0_Scope_Attribute_De…

THINNER COVERAGE BELOW

05 **Hardened JUnit XML intake** IMPROVED 55

JUnit XML ingestion now uses `defusedxml`, rejecting DTD/entity input while retaining the existing size, encoding, structure, and declared-total checks.

— Names library and behavior but no user-facing flag

2026-09-17_CLI_0.6.0_Scope_Attribute_De…

2026-09-17_CLI_0.6.0_Scope_Attribute_Decide NOTES ↗ LATEST STABLE

AI SAFE² CLI 0.6.0 adds `safe2 evidence scope`, `attribute`, and `readiness` commands for deployment-boundary scoping, finding attribution, and release-readiness decisions.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 2026-09-17_CLI_0.6.0_Scope_Attribute_Decide
https://github.com/CyberStrategyInstitute/ai-safe2-framework.git
# already have the repo? check out this version:
$ git checkout 2026-09-17_CLI_0.6.0_Scope_Attribute_Decide
```

Adds `safe2 evidence scope` command to inventory the declared deployment relationship of repository or built-artifact paths as included, excluded, partial, not applicable, or unknown.

Adds `safe2 evidence attribute` command to compare normalized findings across an explicitly trusted baseline and current revision, separating inherited, introduced, changed, resolved, and unknown findings.

Adds `safe2 evidence readiness` command to produce canonical release-readiness JSON and an escaped Markdown card, accepting `--system-identity`, `--assessment-scope`, `--change-attribution`, `--output`, `--card`, and `--strict` options.

Adds exportable JSON schemas `assessment-scope-source-v1`, `change-attribution-source-v1`, and `release-readiness-source-v1` via `safe2 schema export` to anchor evidence contracts.

Introduces three readiness statuses — `hold`, `review`, and `ready_for_human_decision` — with `release_authorized: false` enforced on every result and a named human decision owner required.

> +1 more

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

xAI Grok API

Feedback ⓘ snapshot-20260918 NOTES ↗ RANK

Official SpaceXAI API documentation. Build with Grok: chat, image and video generation, voice, tool calling, and structured outputs.

> Grok API documentation gained a new speech-to-text model option, language-pinned code samples, regional endpoint docs, and a new enterprise Grok Bot management section.

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

01 **Accept-Language header for docs code examples** NEW

80

Adds `Accept-Language` header support to pin API documentation code examples to a specific language and SDK, with supported subtags `python-xai`, `python-openai`, `javascript-ai-sdk`, and `javascript-openai`.

— Exact header and all supported subtags given, directly usable. product docs

02 **grok-voice-transcribe-2.0 speech-to-text model** NEW

60

Adds `grok-voice-transcribe-2.0` as a new speech-to-text model option alongside the existing `grok-voice-transcribe-1.0`; the default model remains `grok-voice-transcribe-1.0`.

— Names both model IDs and default, but no usage mechanism. snapshot-20260918

THINNER COVERAGE BELOW

03 **Regional endpoints documentation** NEW 50

Adds regional endpoints documentation at `https://docs.x.ai/developers/advanced-api-usage/regions.md`.

— Exact doc URL given but no content detail on the endpoints themselves. product docs

04 **Enterprise Grok Bot management docs** NEW 45

Adds enterprise Grok Bot management documentation covering identity and access, private networks, TLS-inspecting proxies, computer management, security, and a security FAQ.

— Lists topic areas but no specific keys, flags or endpoints. product docs

snapshot-20260918 NOTES ↗ LATEST STABLE

> Grok API adds `grok-voice-transcribe-2.0` speech-to-text model alongside the existing v1.0.

WAS THIS USEFUL? 👍 👎

Anthropic

Feedback ⓘ snapshot-20260918 NOTES ↗ RANK

Anthropic provides Claude AI models and APIs for building applications that generate, analyze, and automate work.

> Anthropic extended its Compliance API so Enterprise organizations can retrieve transcripts from Claude in Chrome sessions alongside existing session data.

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

01 Claude in Chrome transcripts in Compliance API NEW

73

Compliance API local session endpoints now return transcripts from Claude in Chrome sessions, identified by `product_surface` value `claude_in_chrome`. Access uses the existing Compliance Access Key and `read:compliance_user_data` scope, and the feature is in beta for Claude Enterprise organizations.

— Names exact field value and scope, but no endpoint path given. snapshot-20260918

> snapshot-20260918 NOTES ↗ LATEST STABLE
>
> Compliance API session endpoints now return Claude in Chrome transcripts for Enterprise orgs

WAS THIS USEFUL? 👍 👎

Google Gemini API

Feedback ⓘ September 17, 2026 NOTES ↗

RANK

Gemini API provides Gemini models for generating text, images, and other multimodal content in applications.

> Gemini API introduced managed Antigravity agents (`antigravity-preview-09-2026`) with stateful, tool-using interaction environments — adding built-in web search, code execution, MCP server integration, function calling, scheduled triggers, network egress controls, streaming, and environment snapshots — while renaming and replacing the agent's file-editing tools in a breaking update ahead of the prior preview's October 5, 2026 shutdown.

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

01 File editing tools renamed and restructured

BREAKING

90

In `antigravity-preview-09-2026` , write_file(path, content) is removed and replaced by write_to_file(TargetFile, CodeContent, Overwrite, Description) for file creation with explicit overwrite control; read_file(path, offset, limit) (byte offsets) is removed and replaced by view_file(AbsolutePath, StartLine, EndLine, ContentOffset) (line-based addressing); list_files(path) is removed and replaced by list_dir(DirectoryPath); and full-file rewrites are replaced by surgical edits via replace_file_content(TargetFile, StartLine, EndLine, TargetContent, ReplacementContent). All built-in tool parameters are renamed from snake_case to PascalCase, so callers parsing `function_call` steps or running tools locally must update parameter names (e.g. `path` becomes `TargetFile` or `AbsolutePath` , `content` becomes `CodeContent`). `antigravity-preview-05-2026` shuts down on October 5, 2026 and all integrations must migrate to `antigravity-preview-09-2026` before that date.

— Exact old/new function signatures and migration deadline given. September 17, 2026

02 Scheduled agent triggers and credential rotation

NEW

87

Adds scheduled Trigger support via `TriggerCreateParams` (with `timeZone` , `displayName` , and an `interactionTemplate`) for unattended recurring agent runs, plus full lifecycle management: listing triggers, getting trigger details including `nextRunTime` , pausing/resuming via `TriggerUpdateStatus` (`PAUSED` / `ACTIVE`), and deleting triggers. `listTriggerExecutions` retrieves past runs, each exposing `startTime` , `endTime` , and a linked `interactionId` for result inspection via `GetInteractionByIdRequest` . Stored `credential` references (e.g. `'github-`

`production'`) can be attached to triggers and egress allowlists so secrets can be rotated without modifying trigger definitions.

— Comprehensive lifecycle and field names, no runnable snippet. product docs

03 **Environment snapshot download endpoint** NEW 78

Adds environment snapshot download via `GET https://generativelanguage.googleapis.com/v1beta/files/environment-<ENVIRONMENT_ID>:download?alt=media`, returning a `snapshot.tar` archive.

— Exact endpoint and response format, immediately runnable. product docs

04 **Antigravity agent model and interaction environments** NEW 76

Introduces the `antigravity-preview-09-2026` agent model as a new agentic interaction surface in the Gemini API, provisioned via `CreateAgentInteractionEnvironment` and run via `CreateAgentInteraction` / `AgentOption`, each returning an `environmentId` and `interactionId` per session. Agents are configured with `Agent`, `AgentConfig`, and `AntigravityAgentConfig`, supporting `systemInstruction` and `baseEnvironment` set against `base_agent: antigravity-preview-09-2026`.

— Named core APIs and fields but no runnable usage example. product docs

05 **New file and content search tools** NEW 73

Adds find_by_name(SearchDirectory, Pattern, MaxDepth) for file and directory search and grep_search(SearchPath, Query, IsRegex) for code/content search with regex support — capabilities the previous agent lacked, relying on shell commands instead.

— Named function signatures, no worked call example. September 17, 2026

06 **Custom function calling with REQUIRES_ACTION** NEW 71

Adds custom function-calling support: agents can return a `REQUIRES_ACTION` status with `FunctionCallStep` items, and callers submit results back via `FunctionResultStep` and `FunctionResultStepResultUnion` in a follow-up interaction using `previousInteractionId` and `environmentId`.

— Full request/response flow named, no code sample. product docs

07 **Token budget control and multi-turn resumption** NEW 70

Adds `AntigravityAgentConfig` with a `maxTotalTokens` field to cap total token consumption per interaction, with the agent returning an `incomplete` status when the budget is hit. A stopped interaction can be resumed by passing its `interactionId` as `previousInteractionId` along with a new `maxTotalTokens` budget, and agents

more generally support multi-turn conversations via `previousInteractionId` to continue from a prior interaction's state.

— Mechanism and fields named, no worked example. product docs

08 **Built-in agent tools: search, URL context, code execution, MCP**

NEW

68

Adds `GoogleSearch` and `URLContext` as built-in tools configurable on `CreateAgentInteractionEnvironment` for grounded web search and URL-based context, `CodeExecution` for writing and running code inside the interaction environment, and `MCPServer` for registering remote HTTP MCP servers as agent tools.

— Four named tools but no config example shown. product docs

THINNER COVERAGE BELOW

09 **Asynchronous background execution with polling** NEW

55

Supports asynchronous (background) interaction execution with polling via `GetInteractionByIdRequest`, gated on `InteractionStatus` values `IN_PROGRESS` and `COMPLETED`.

— Named request and status values, minimal mechanism detail. product docs

10 **Multimodal image and text inputs** NEW

52

Supports multimodal (image + text) inputs to the agent via `ImageContent` and `TextContent` content types, with `ImageContentMimeType` for specifying formats such as `IMAGE_PNG`.

— Content types named but no sample payload. product docs

11 **Inline and repository sources for agent environments** NEW

52

Adds `Source` and `SourceType` for supplying context into agent environments: `INLINE` for supplying documents/code directly on the `Environment`, and `REPOSITORY` for pulling in a repository source, both settable at environment creation time.

— Named enum values, no example of use. product docs

12 **Network egress control for agent environments** NEW

45

Adds network egress control on the agent environment via `EnvironmentNetworkEgressAllowlist`, `Allowlist`, `AllowlistEntry`, and `Transform` for outbound traffic policy.

— Named types only, mechanism unexplained. product docs

13 **Real-time streaming of interaction events** NEW 43

Adds `EventStream<InteractionSSEStreamEvent>` and `StepStop` for streaming agent interaction events in real time.

— Only type names given, no behaviour detail. product docs

└──▷ **BREAKING ON UPGRADE**

! All built-in tool parameters are renamed from snake_case to PascalCase in `antigravity-preview-09-2026`; callers parsing `function_call` steps or running tools locally must update parameter names (e.g., `path` becomes `TargetFile` or `AbsolutePath`, `content` becomes `CodeContent`, etc.).

! write_file(path, content) is removed; file creation now requires write_to_file(TargetFile, CodeContent, Overwrite, Description) and file editing now requires replace_file_content(TargetFile, StartLine, EndLine, TargetContent, ReplacementContent).

! read_file(path, offset, limit) (byte offsets) is removed and replaced by view_file(AbsolutePath, StartLine, EndLine, ContentOffset) (line-based addressing).

! list_files(path) is removed and replaced by list_dir(DirectoryPath).

! `antigravity-preview-05-2026` shuts down on October 5, 2026; all integrations must migrate to `antigravity-preview-09-2026` before that date.

September 17, 2026 NOTES ↗ LATEST STABLE

Antigravity Agent 09-2026 replaces the May preview with new built-in tools using PascalCase parameters and line-range file editing.

Adds write_to_file(TargetFile, CodeContent, Overwrite, Description) to replace the `write_file` function for file creation, with explicit overwrite control and a description field.

Adds replace_file_content(TargetFile, StartLine, EndLine, TargetContent, ReplacementContent) for surgical line-range file edits instead of full rewrites.

Adds view_file(AbsolutePath, StartLine, EndLine, ContentOffset) to replace `read_file`, switching from byte offsets to line-based addressing.

Adds list_dir(DirectoryPath) to replace `list_files` for directory listing.

Adds find_by_name(SearchDirectory, Pattern, MaxDepth) for file and directory search, a capability the previous agent lacked (relying on shell commands instead).

+1 more

└──▷ **BREAKING ON UPGRADE**

! All built-in tool parameters are renamed from snake_case to PascalCase in `antigravity-preview-09-2026`; callers parsing `function_call` steps or running tools locally must update parameter names (e.g., `path` becomes `TargetFile` or `AbsolutePath`, `content` becomes `CodeContent`, etc.).

! write_file(path, content) is removed; file creation now requires write_to_file(TargetFile, CodeContent, Overwrite, Description) and file editing now requires replace_file_content(TargetFile, StartLine, EndLine, TargetContent, ReplacementContent).

! read_file(path, offset, limit) (byte offsets) is removed and replaced by view_file(AbsolutePath, StartLine, EndLine, ContentOffset) (line-based addressing).

! list_files(path) is removed and replaced by list_dir(DirectoryPath).

! `antigravity-preview-05-2026` shuts down on October 5, 2026; all integrations must migrate to `antigravity-preview-09-2026` before that date.

WAS THIS USEFUL?  

◆ Open-Weight Models

Alibaba Qwen Models

Feedback ⓘ qwen3.8-omni-flash NOTES ↗

RANK

Qwen is Alibaba's language and multimodal model family, including coding, vision, image, speech, and reasoning models.

> Alibaba released Qwen3.8-Omni-Flash, a new multimodal model adding image, audio and video understanding alongside flexible reasoning modes, agentic tooling and web search.

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

01 **Qwen3.8-Omni-Flash multimodal model release** NEW 46

New model `qwen3.8-omni-flash` supports text, image, audio, and video inputs with text output for multimodal analysis and agent applications. It offers thinking and non-thinking modes for flexible reasoning control, function calling for agentic workflows, web search integration, and context caching for audio and video understanding.

— Lists capabilities but no API params, limits, or usage steps. qwen3.8-omni-flash

> qwen3.8-omni-flash NOTES ↗ LATEST STABLE
>
> Qwen3.8-Omni-Flash adds multimodal input (text, image, audio, video) with thinking/non-thinking modes, function calling, and web search.
>
> Supports text, image, audio, and video inputs with text output for multimodal analysis and agent applications.
>
> Supports thinking and non-thinking modes for flexible reasoning control.
>
> Supports function calling for agentic workflows.
>
> Supports web search integration.
>
> Supports context caching for audio and video understanding.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

▷ Subscribe Feedback on this issue