THE AI TOOLCHAIN

NO. 077

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 29, 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 183 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 43 of 44 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. Mastra lets you swap the model behind an agent as an experiment without editing the agent definition. puts a human approval gate in front of risky MCP tool calls and

vim ? help

logs the findings. And Unsloth trains several LoRA adapters in one batched run on shared base weights instead of one job per adapter.

EVALUATE

Compare a new model on a live agent without forking the agent

Mastra's agent experiments bind a different model to an existing agent at run time, so the usual copy-the-agent-and-edit-the-config step before an A/B goes away. Braintrust's one-line Cursor session tracing closes the other half: coding-agent runs land as traces you can score instead of scrollback you lose.

Mastra · Braintrust

GOVERN

Hold a risky MCP tool call for a human before it executes

Gram's Guardrails now pause a tool call for approval and record the decision in a risk-findings dataset, which turns 'the agent did something it shouldn't' into a reviewable queue and an audit trail. LanceDB moves in the same direction on the data side, putting embedding models and custom endpoints behind an authorization API so an arbitrary outbound endpoint isn't a config field anyone can set.

gram · LanceDB

BUILD

Train many LoRA adapters in one run instead of one job per adapter

Unsloth's multi-LoRA batched fine-tuning with new Triton kernels shares the frozen base weights across adapters in a single pass, so per-customer or per-task adapters stop costing a separate GPU job each. Docling's automatic OCR recovery catches PDFs whose text layer is broken — the ones that silently produce empty chunks in an ingestion pipeline — and re-reads them instead.

Unsloth · Docling

BUILD

◆ Model Training & Fine-Tuning

PyTorch

Feedback ⓘ announced · Oct 29, 2026 BLOG ↗ RANK

Tensors and Dynamic neural networks in Python with strong GPU acceleration

> PyTorch 2.15 replaces the default distributed NCCL backend with a rewritten implementation and adds fine-grained FSDP2 gradient-precision controls, while expanding hardware support with ROCm 10.0 wheels, a new FlyDSL kernel DSL on AMD gfx950, Apple Silicon FP8 matmul, CUDA 13.2/13.4 wheels, and Python 3.15 support, alongside several breaking changes to checkpoint loading, custom-op exception handling, and activation checkpointing.

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

01 Rewritten default NCCL backend

BREAKING

95

PyTorch 2.15 replaces the default NCCL backend with a rewritten implementation featuring a hang-dump flight recorder; `backend='nccl'` now creates this new implementation. Roll back to the previous implementation with `TORCH_DIST_USE_NCCL2=0` or `backend='nccl-legacy'` (available for this release only, removed next release), or opt into independent per-peer P2P communicators with `backend='nccl-lazy'`.

Keep a distributed training job on the old NCCL backend while validating the new default in staging.

```
$ TORCH_DIST_USE_NCCL2=0 torchrun --nproc_per_node=8 train.py
```

Initialize a process group with the new NCCL backend and opt into lazy independent P2P communicators for overlapping concurrent sends to different peers.

```python
import torch
import torch.distributed as dist

# Eager communicator creation, one device per group
dist.init_process_group('nccl', device_id=torch.device('cuda',
local_rank))

# Or: independent P2P communicators per peer pair
dist.init_process_group('nccl-lazy',
device_id=torch.device('cuda', local_rank))
```

— Named flags and runnable rollback commands given, full mechanism described.

launch-20261029-bfb5cbc8 v2.15.0

02 **FSDP2 gradient precision controls** BREAKING 95

FSDP2 now supports fp32 gradient accumulation and reduction for precision-sensitive parameters via `MixedPrecisionPolicy`'s `param_dtype_override_fn`, CUDA-graph-captured gradient accumulation, and gradient reduction overlapped with compute. With `reduce_dtype=None`, gradients now accumulate directly in `.grad` following each parameter's `grad_dtype` instead of the previous `param_dtype` default; set `reduce_dtype` explicitly to restore 2.14 behavior.

Keep MoE router gates and norm weights in fp32 while training the rest of the model in bf16 with FSDP2.

python

```python
from torch.distributed.fsdp import MixedPrecisionPolicy,
fully_shard
import torch

def keep_fp32(param_name, param):
    return 'gate' in param_name or 'norm' in param_name

policy = MixedPrecisionPolicy(
    param_dtype=torch.bfloat16,
    param_dtype_override_fn=keep_fp32,
)
fully_shard(model, mp_policy=policy)
```

Keep fp32 precision for FSDP2 gradient reduction when training with bf16 compute by setting `reduce_dtype` explicitly on `MixedPrecisionPolicy`.

```python
from torch.distributed.fsdp import fully_shard,
MixedPrecisionPolicy
import torch

# Gradients accumulate and reduce-scatter in fp32 (new 2.15
default)
mp_policy = MixedPrecisionPolicy(param_dtype=torch.bfloat16)
fully_shard(model, mp_policy=mp_policy)

# To keep bf16 reduction (2.14 behaviour), set reduce_dtype
explicitly
mp_policy_bf16 = MixedPrecisionPolicy(param_dtype=torch.bfloat16,
reduce_dtype=torch.bfloat16)
fully_shard(model, mp_policy=mp_policy_bf16)
```

— Full code examples and exact behavior change documented. launch-20261029-bfb5cbc8 v2.15.0

03 **Apple Silicon MPS FP8 and grouped_mm** NEW 90

Apple Silicon now supports `float8_e4m3fn`, `_scaled_mm`, a new `grouped_mm` with a decode fast path (1.32–1.34x tokens/s on MoE decode), and public `torch.mps.Stream` APIs.

Run FP8 matrix multiplication on Apple Silicon using the new MPS float8 support.

```python
import torch

x = torch.randn(128, 256, device='mps').to(torch.float8_e4m3fn)
w = torch.randn(512, 256, device='mps').to(torch.float8_e4m3fn)
scale_x = torch.tensor(1.0, device='mps')
scale_w = torch.tensor(1.0, device='mps')
out = torch._scaled_mm(x, w.t(), scale_a=scale_x,
scale_b=scale_w)
```

— Runnable example plus concrete speedup numbers given. launch-20261029-bfb5cbc8 v2.15.0

04 **Stricter weights_only default for nested tensors** BREAKING 80

Nested jagged and strided nested tensors are removed from the default `weights_only=True` allowlist; loading checkpoints containing them now raises

`Unsupported global` unless you use `torch.serialization.safe_globals` or `weights_only=False`.

Load a checkpoint that contains nested tensors under the new strict `weights_only=True` default by explicitly allowlisting the required globals.

```python
import torch
from torch.nested._internal.nested_tensor import _rebuild_njt,
NestedTensor

with torch.serialization.safe_globals([_rebuild_njt,
NestedTensor]):
    loaded = torch.load('njt.pt', weights_only=True)
```

— Exact error and workaround with runnable code given. v2.15.0

05 **Pipeline parallelism send budgeting and per-edge P2P** BREAKING

75

Pipeline parallelism now accepts a `max_outstanding_sends` budget and staggers FSDP unshards across each rank's startup bubble instead of issuing all at the start of a step. It also gains per-directed-edge P2P communicators, enabled via `dist_config.pipeline_per_edge_p2p = True` or `TORCH_DISTRIBUTED_PIPELINE_PER_EDGE_P2P=1` — renamed from `pipeline_per_direction_p2p` / `TORCH_DISTRIBUTED_PIPELINE_PER_DIRECTION_P2P`, whose old attribute now raises `AttributeError` and old env var is silently ignored.

— Config keys named but no runnable example provided. launch-20261029-bfb5cbc8 v2.15.0

06 **AMD gfx1250/ROCm 10.0 hardware support** NEW

75

Inductor now uses the Tensor Data Mover on AMD gfx1250 (MI450) for dense `mm` / `addmm`, pointwise and reduction codegen, and FlexAttention forward and decode; gfx1250 is added to the nightly wheel architecture list. PyTorch 2.15 also adds ROCm 10.0 wheels built on AMD's TheRock packages, with ROCm installed as a pip dependency (the matching `rocm` package) rather than bundled libraries.

— Concrete hardware targets and packaging change named, no example.

launch-20261029-bfb5cbc8 v2.15.0

07 **CUDA 13.2/13.4 wheels with Rubin support** NEW

60

The default PyTorch wheel now uses CUDA 13.2 (13.2.2 toolkit), with CUDA 13.4 GA wheels available alongside it for Linux x86_64, Linux aarch64, and Windows, including experimental support for NVIDIA Rubin GPUs.

— Specific CUDA versions and platforms, no install command given.

launch-20261029-bfb5cbc8 v2.15.0

08 **Selective activation checkpointing policy change** BREAKING 60

Selective activation checkpointing now saves (rather than recomputes) outputs of operators with an ordered effect when the policy returns `CheckpointPolicy.PREFER_RECOMPUTE`; use `CheckpointPolicy.MUST_RECOMPUTE` to keep the 2.14 behaviour.

— Exact policy constants named for before and after. v2.15.0

THINNER COVERAGE BELOW

09 **Native exception/warning translation for custom ops** BREAKING 55

Operators called through `torch.ops.*` and `torch._C._dispatch_call_boxed` now use PyTorch's own exception and warning translation, so user-defined pybind11 exception translation is lost and `TORCH_WARN` warnings now reach Python's `warnings` module.

— Named APIs affected, no migration example given. v2.15.0

10 **Distributed checkpoint metadata now includes strides** BREAKING 55

Distributed checkpoint `.metadata` files now include tensor strides (`TensorProperties.strides`), so checkpoints saved with 2.15 cannot be loaded by PyTorch 2.14 or earlier, raising CheckpointException: too many values to unpack (expected 5).

— Exact error message given, no migration path shown. v2.15.0

11 **FlyDSL joins kernel DSL ecosystem** NEW 50

FlyDSL joins CuTeDSL, Triton, and Helion as a kernel DSL; on AMD gfx950, Inductor now autotunes FlyDSL GEMM and `grouped_mm` templates (requires FlyDSL 0.3.x).

— Named DSLs and version requirement, no usage example. launch-20261029-bfb5cbc8 v2.15.0

12 **Python 3.15 and free-threaded build support** NEW 50

PyTorch now supports Python 3.11 through 3.15, including free-threaded `3.14t` and `3.15t` builds, with Dynamo adding `torch.compile` support for Python 3.15.

— Named versions, no example of free-threaded usage. launch-20261029-bfb5cbc8 v2.15.0

13 **Stricter re-import error handling** BREAKING 45

importlib.reload(torch) and re-importing `torch` after a failed import now raise `ImportError` instead of producing misleading errors or segfaults.

— Clear before/after behavior, no workaround needed. v2.15.0

14 **NIXL one-sided transport backend** NEW 40

`torch.distributed` now adds one-sided transports via a NIXL backend supporting CUDA-graph-capturable transfers.

— Only prose description, no example or config shown. v2.15.0

15 **Faster NVFP4 decoding on Blackwell** IMPROVED 35

NVIDIA Blackwell GPU LLM decoding at 4-bit (NVFP4) precision is now faster through fewer, better-overlapped NVGEMM kernel launches.

— Brief prose only, no numbers or example given. v2.15.0

16 **Intel GPU SYCL authoring and profiling** IMPROVED 25

Intel GPU support gains simpler SYCL kernel authoring and multi-GPU profiling.

— Vague description, no specifics or example provided. v2.15.0

└──▷ BREAKING ON UPGRADE

! FSDP2 `reduce_dtype=None` now causes gradients to accumulate directly in `.grad` following each parameter's `grad_dtype`, changing the previous default behavior.

! importlib.reload(torch) and re-importing `torch` after a failed import now raise `ImportError` instead of producing misleading errors or segfaults.

! Nested jagged and strided nested tensors are removed from the default `weights_only=True` allowlist; loading checkpoints containing them now raises `Unsupported global` unless you use `torch.serialization.safe_globals` or `weights_only=False`.

! Operators called through `torch.ops.*` and `torch._C._dispatch_call_boxed` now use PyTorch's own exception and warning translation, so user-defined pybind11 exception translation is lost and `TORCH_WARN` warnings now reach Python's `warnings` module.

! Selective activation checkpointing now saves (rather than recomputes) outputs of operators with an ordered effect when the policy returns `CheckpointPolicy.PREFER_RECOMPUTE`; use `CheckpointPolicy.MUST_RECOMPUTE` to keep the 2.14 behaviour.

! `backend='nccl'` now creates the rewritten NCCL implementation; the legacy backend is available as `backend='nccl-legacy'` or `TORCH_DIST_USE_NCCL2=0` for this release only — both will be removed in the next release.

! Distributed checkpoint `.metadata` files now include tensor strides (`TensorProperties.strides`), so checkpoints saved with 2.15 cannot be loaded by PyTorch 2.14 or earlier (CheckpointException: too many values to unpack (expected 5)).

! With MixedPrecisionPolicy(reduce_dtype=None), `fully_shard` now accumulates and reduces gradients in the parameter's original dtype instead of `param_dtype`; set `reduce_dtype` explicitly to restore 2.14 behaviour.

! `torch.distributed.config.pipeline_per_direction_p2p` is renamed to `pipeline_per_edge_p2p`; the old attribute raises `AttributeError` and `TORCH_DISTRIBUTED_PIPELINE_PER_DIRECTION_P2P` is silently ignored.

WAS THIS USEFUL? 👍 👎

◆ AI Coding Agents

SST OpenCode

Feedback ⓘ changelog · Oct 29, 2026 CHANGELOG ↗

RANK

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

> OpenCode's latest release adds an experimental session media API with content-addressed blob storage and experimental Desktop voice calls, alongside smaller API and CLI refinements, while also shipping several breaking changes: a restructured `@opencode/ai` package, removal of the GitHub Action, and a required OpenTUI 0.7.0 upgrade for TUI plugins.

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

01 @opencode/ai package Promise/Effect API restructuring

95

BREAKING

The `@opencode/ai` package root now exports the Promise API; the Effect API moved to `@opencode/ai/effect`. Replace `import { AI } from '@opencode/ai/promise'` with `import { AI } from '@opencode/ai'` and `import { LLM } from '@opencode/ai'` with `import { LLM } from '@opencode/ai/effect'`. The `@opencode/ai/promise` subpath and the wildcard subpath export are removed; use the explicit provider and protocol entry points.

– Exact import paths and migration steps given changelog-20261029-2df495f7

02 TUI plugins require OpenTUI 0.7.0

BREAKING

90

TUI plugins now require `@opentui/core` and `@opentui/solid` 0.7.0 (up from 0.6.8); plugins depending on OpenTUI 0.6 fail to install until upgraded. In OpenTUI 0.7, custom renderables must assign `renderBefore`, `renderAfter`, and `onLifecyclePass` in the constructor instead of as class fields, call refreshHooks() after adding a hook by reflection, and replace defineNativeIntegration() with a `static nativeIntegration` declaration.

– Exact version numbers and concrete code migration steps changelog-20261029-2df495f7

03 Experimental session media API with content-addressed blob storage

NEW

78

Adds an experimental API for session media with opt-in media URI support, and enables experimental content-addressed blob storage for session media via the `experimental.media_blob` config key.

Enable experimental content-addressed blob storage so session media is deduplicated and addressed by content hash.

```yaml
experimental:
  media_blob: true
```

— Named config key and usage example, but marked experimental with limited mechanism detail

changelog-20261029-2df495f7

04 **CLI flag handling and JSON error output** IMPROVED 65

`opencode run --format json` now returns server connection errors as structured JSON instead of unstructured output, and the `--server` and `--standalone` flags can now appear anywhere in supported CLI commands rather than only at the start.

Get structured JSON output from a failed server connection when running OpenCode in CI or automation pipelines.

```
$ opencode run --format json --server
```

— Named flags and one example, but limited mechanism beyond that

changelog-20261029-2df495f7

05 **Machine hostname exposed via /api/info** IMPROVED 60

The `/api/info` endpoint now includes the machine hostname in its response, useful for identifying which host is serving in multi-node deployments.

Retrieve the machine hostname and other server metadata — useful for multi-node deployments that need to confirm which host is serving.

```
$ curl http://localhost:<port>/api/info
```

— Named endpoint with example, but minimal further detail changelog-20261029-2df495f7

THINNER COVERAGE BELOW

06 **OpenCode GitHub Action removed** DEPRECATED 50

The OpenCode GitHub Action is no longer published. Remove it from your workflows.

— Clear instruction but no detail on replacement or cause changelog-20261029-2df495f7

07 **Type filtering for model-list API and plugin events** NEW 37

Model-list API requests can now filter by type, and plugins can filter event subscriptions by type.

— Thin one-line description with no parameter names or examples

changelog-20261029-2df495f7

08 **MCP server titles exposed from serverInfo** NEW 35

MCP server titles are now exposed from `serverInfo`.

— Single-line addition naming a field but no further mechanism

changelog-20261029-2df495f7

09 **Experimental voice calls in Desktop beta** NEW 25

Adds experimental Console voice calls in the Desktop beta.

— Bare mention with no mechanism or usage detail changelog-20261029-2df495f7

└──▷ **BREAKING ON UPGRADE**

! The `@opencode/ai` package root now exports the Promise API; the Effect API moved to `@opencode/ai/effect`. Replace `import { AI } from '@opencode/ai/promise'` with `import { AI } from '@opencode/ai'` and `import { LLM } from '@opencode/ai'` with `import { LLM } from '@opencode/ai/effect'`. The `@opencode/ai/promise` subpath and the wildcard subpath export are removed; use the explicit provider and protocol entry points.

! The OpenCode GitHub Action is no longer published. Remove it from your workflows.

! TUI plugins now require `@opentui/core` and `@opentui/solid` 0.7.0 (up from 0.6.8). Plugins depending on OpenTUI 0.6 fail to install until upgraded. In OpenTUI 0.7, custom renderables must assign `renderBefore`, `renderAfter`, and `onLifecyclePass` in the constructor instead of as class fields, call refreshHooks() after adding a hook by reflection, and replace defineNativeIntegration() with a `static nativeIntegration` declaration.

WAS THIS USEFUL? 👍 👎

Warp

Feedback ⓘ 2026.10.22 (v0.2026.10.21.08.25) NOTES ↗ RANK

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

> Warp focused on cloud-agent session reliability this cycle—reusing pre-warmed sandboxes, swapping in correct Git identities, and aligning attachment handling for ACP agents—while also introducing Warp Factories, a YAML-defined infrastructure for building software factories.

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

01 Warp Factories for software development workflows NEW

55

Warp introduces Factories, open infrastructure for building a software factory. An Activity view groups tasks into Triage, Planning, and Building stages, with linked issues, implementation plans, channels, and Needs attention statuses. Factories can be defined as code via a YAML configuration specifying repositories, agent models and roles, and triggers such as a GitHub pull-request trigger.

⋮

— Described via screenshots only, no commands or config keys given

2026.10.22 (v0.2026.10.21.08.25)

02 Pre-warmed cloud agents adopt run author identity IMPROVED

50

Pre-warmed cloud agents now replace standby Git credentials and use the assigned run's author identity while preserving setup overrides.

— Explains mechanism but names no flags or config surfaces

2026.10.22 (v0.2026.10.21.08.25)

03 Resource-link attachments for ACP agents IMPROVED

45

Initial file attachments are now sent as resource links to ACP agents, matching the behaviour already used for follow-up attachments.

— Names ACP agents but no API or config detail 2026.10.22 (v0.2026.10.21.08.25)

04 Resumed runs reuse pre-warmed sandboxes IMPROVED

38

Resumed agent runs can now reuse pre-warmed sandboxes while preserving conversation history and queued follow-ups.

— Brief behavioural note with no named surfaces 2026.10.22 (v0.2026.10.21.08.25)

WAS THIS USEFUL? 👍 👎

mex

Feedback ⓘ v0.8.5 CODE ↗ RANK

Mex indexes codebases through a code graph and supplies a repo-local Markdown wiki to AI coding agents.

> Mex's headline addition is a one-paste agent handoff flow (`mex setup next --start`) and a new TLA+/TLC-checked statechart skill for explaining flows and recovery paths, alongside tighter setup orchestration, documented CLI flag constraints, and a new Node.js version floor.

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

01 **One-paste agent setup handoff** NEW 85

Running `mex setup next --start` gives agents that mex cannot launch directly a one-paste setup flow: it hands out parallel task files, applies ready batches, and continues a saved plan across sessions.

Hand off setup to an agent mex cannot launch directly — paste this into the agent chat to start a parallel, resumable setup flow.

```
$ mex setup next --start
```

▶ View command output v0.8.5

— Exact runnable command with clear mechanism described. v0.8.5

02 **Documented setup CLI mode constraints** IMPROVED 80

`mex setup --help` now describes terminal and browser modes and documents which options `--browser` cannot be combined with (`--cli`, `--tool`, `--yes`, `--dry-run`), as well as what `--no-open` does.

— Names every flag and an exact help command to check them. v0.8.5

03 **Statechart skill with TLA+/TLC verification** NEW 75

The new `mex-statechart` skill explains flows, bugs and recovery paths using TLA+/TLC-checked state diagrams, with optional Hub previews for interactive state graph and path inspection, and an explicit Save that writes an immutable Markdown-owned snapshot.

— Mechanism and verification method named, but no invocation example. v0.8.5

04 **Setup population: ranking, token usage, more agents** IMPROVED 75

Setup now ranks subsystem candidates by indexed file count, recent Git activity, and fan-in, caps slots per area, and sizes the coverage target automatically. It also reports each agent's token usage and its usage limit during population, and launches background Cursor, GitHub Copilot and OpenCode CLI sessions alongside Claude Code and Codex.

— Several concrete ranking signals and agent names, but no new command. v0.8.5

05 **Node.js 22.5 minimum enforced at startup** BREAKING 60

mex now exits with code 1 before any command runs when the Node.js version is older than 22.5, naming both the required and the running versions.

— Exact version threshold and exit behavior named, no migration steps. v0.8.5

THINNER COVERAGE BELOW

06 **Wiki apply batching and auto-structure** IMPROVED 55

Wiki apply now batches writes, names code by file, and applies automatic structure; files created by wiki apply receive a name, description and date.

— Describes behavior change but no config or command to invoke it. v0.8.5

07 **Hub hosted-beta interest form** NEW 40

The Hub now offers an opt-in hosted-beta interest form accessible from setup, Relays and Overview.

— UI locations named but no further mechanism given. v0.8.5

08 **Node naming, fingerprints and aliases in authoring** NEW 35

Authoring now lets agents name nodes while mex records fingerprints, and it supports aliases.

— Bare description of capability without mechanism or example. v0.8.5

WAS THIS USEFUL?  

Sourcegraph Amp

Feedback ⓘ macOS Satellites (2026-10-29)

BLOG ↗ RANK

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

> Amp introduced macOS Satellites, cloud-hosted Macs that let the agent build and test iOS/macOS apps directly from a thread, and opened up self-hosted orb providers to individual users without workspace admin access.

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

01 macOS Satellites for building and testing Apple apps NEW

84

Amp automatically provisions a cloud Mac with Xcode and simulators in about 30 seconds when a thread requires macOS, streams the Mac's screen live in the thread's Mac tab, and keeps it running for 20 minutes after finishing so follow-up messages reuse the same build cache. Users can request a larger `mac.large` instance (12 CPUs, 28 GB memory at $4/hour) or disable the default `mac.medium` under Mac Satellites in Settings → Experimental Features, where the feature is currently in beta and joined via a waitlist advanced by daily clicks on Move Me Up .





— Names mechanism, timing, instance types and exact settings path.

macOS Satellites (2026-10-29)

THINNER COVERAGE BELOW

02 Personal self-hosted orb providers NEW

56

Any user can now add a self-hosted orb provider under personal Settings → Orbs without requiring workspace admin access; these providers appear under Self-Hosted marked Personal when creating a thread.

— Names exact settings path and UI label but no further mechanism.

Your Own Self-Hosted Orbs (2026-10-28)

WAS THIS USEFUL? 👍 👎

Block Goose

Feedback ⓘ v1.57.0 NOTES ↗ CODE ↗ RANK

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

> Goose tightened its security posture this release, hardening allowlist enforcement for outbound HTTP requests and letting external callers supply their own session IDs.

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

01 **Caller-supplied Goose session IDs** NEW 45

Callers can now supply their own Goose session IDs when starting a session, enabling deterministic session tracking from external systems.

— Describes capability but no API/flag name given v1.57.0

02 **Stricter HTTPS allowlist enforcement** IMPROVED 45

Allowlist enforcement now requires HTTPS, validates redirects, and blocks requests on policy failure rather than silently allowing them.

— Explains before/after behavior but no config surface named v1.57.0

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.20.0 NOTES ↗

RANK

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

> Zeroshot's v10.20.0 release lets hosts inject per-agent execution context into its agent loops and hardens the reliability of its Git credential handling and Restic backup streaming.

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

01 **Host-supplied agent context via AgentContextProvider** NEW

68

Hosts can supply per-agent execution context to Codex, Claude, Copilot, and Pi through `AgentContextProvider`, with fresh context delivered to reused sessions, resumed attempts, and auxiliary agents.

— Names the API surface and agents but no usage example v10.20.0

THINNER COVERAGE BELOW

02 **Bounded-memory Restic backup streaming** IMPROVED 48

Restic backup streaming now drains complete stdout/stderr with bounded memory, retains only the latest backup summary, and preserves the previously published recovery point on failure.

— Explains before/after behavior but no runnable surface v10.20.0

03 **Hardened Git credential handling** IMPROVED 45

Git credential handling now follows the native transport's credential-selection policy and redacts token, password, and API-key values while preserving private protocol stdout.

— Describes mechanism but no config surface or command to act on v10.20.0

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ desktop-v0.26.3 NOTES ↗

RANK

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

> Qwen Code's biggest addition this window is persistent Workflow mode (`--workflow-mode on`) paired with a new offline `qwen-managed-agent-admin` launcher, while the TypeScript SDK gained session resume, multi-modal input, an `export` command, and two breaking changes (`disable*` → `enable*` settings rename, removal of `read_many_files`); the desktop build added workspace creation from Git URLs, a headless shared agent loop, and several thin managed-agent and core internals improvements.

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

01 Persistent Workflow mode NEW

90

Workflow mode can now persist for a session using `--workflow-mode on` on the CLI, `/workflows mode on` in-session, or via SDK and ACP controls, with the saved mode shown in Ink and OpenTUI.

Keep Workflow mode active for a long-running session so all tasks run as persistent workflows without re-enabling each time.

```
$ qwen --workflow-mode on
```

— Exact flag, in-session command, and runnable example given. v0.26.3

02 qwen-managed-agent-admin maintenance launcher NEW

80

The new `qwen-managed-agent-admin` launcher runs offline managed-agent maintenance commands by name, with built-in help and unknown-command handling.

Run offline managed-agent maintenance without starting a full agent session.

```
$ qwen-managed-agent-admin --help
```

— Runnable command with help flag shown. v0.26.3

03 Settings keys renamed from disable* to enable* BREAKING

70

Settings keys previously named with `disable*` prefixes are renamed to `enable*` prefixes for a positive naming convention; any configuration relying on the old `disable*` key names will break.

— Names exact prefix change; no full key mapping list. sdk-typescript-v0.1.25

04 **read_many_files tool removed** BREAKING 70

The `read_many_files` tool is removed; its functionality moves to a `readManyFiles` utility available only via user `@-commands`.

— Names old and new surfaces with migration path. sdk-typescript-v0.1.25

05 **Workspace creation and management in web shell** NEW 70

Workspaces can now be created directly from a Git URL in the web shell, files can be uploaded to managed workspaces from the agent interface, extra workspace directories can be managed from the settings panel in the web shell, and external workspace storage ownership can be registered so the managed agent tracks which workspaces belong to external providers.

— Names UI paths but no exact navigation steps. desktop-v0.26.3

06 **Session resume via SDK query options** NEW 65

Previous sessions can be resumed by passing a session ID, using the new `resume` and `continue` options added to the SDK query interface.

— Names exact options but no full code sample. sdk-typescript-v0.1.25

07 **Export session history command** NEW 65

A new `export` command exports session history to markdown or HTML formats.

— Names command and formats, no exact syntax shown. sdk-typescript-v0.1.25

08 **Compact Web Shell input via composerVariant simple** NEW 65

A new `composerVariant simple` setting opts into a compact Web Shell input with a limited add menu and shared editor.

— Names exact setting value, no file path given. v0.26.3

09 **Generated UI persistence and continuation** IMPROVED 65

Generated UI cards let users continue analysis with reviewed parameters, adjusting and rerunning without restarting; cards now also persist inputs and state across history navigation, session switching, and split-view changes within the same page. Generated

UI is offered only to sessions whose client can render MCP Apps, with unsupported clients receiving a clear unsupported-client reason.

– Explains behavior and scope but no config key. v0.26.3 desktop-v0.26.3

10 **Archived conversations management in Web Shell** NEW 60

Archived conversations can now be managed in Web Shell under Settings → Archived conversations, with archive groups, filters, and lifecycle actions.

– Exact UI navigation path given. v0.26.3

11 **Failures ledger for /verify-pr** IMPROVED 60

`/verify-pr` now uses a failures ledger that rejects findings or blocked verdicts unless failures are attributed to the PR.

– Names command and behavior, no exact ledger format shown. v0.26.3

12 **Managed Agent embedding in Java host** NEW 60

Managed Agent opt-in can now be embedded into a Java host's JVM and Spring context, with dedicated connection pooling or a supplied DataSource.

– Names Spring context and DataSource option, no exact API. v0.26.3

THINNER COVERAGE BELOW

13 **FORK_MODE support for ProcessTransport** NEW 55

`ProcessTransport` now supports `FORK_MODE` for Electron IPC integration.

– Names exact flag and transport but limited mechanism detail. sdk-typescript-v0.1.25

14 **Debug logging routed to logfile-first debugLogger** IMPROVED 50

Debug mode now routes all console calls to a logfile-first `debugLogger` instead of stdout.

– Names mechanism but no file path or flag. sdk-typescript-v0.1.25

15 **Multi-modal input across content generators** NEW 50

Qwen Code adds multi-modal support for image, PDF, and audio inputs across all content generators.

– Names input types but no API or flag. sdk-typescript-v0.1.25

16 **Skip redundant UserPromptSubmit hook context** IMPROVED 50

Qwen Code now skips re-injecting an unchanged `UserPromptSubmit` hook context block to reduce redundant prompt overhead.

— Names exact hook block but limited further mechanism. desktop-v0.26.3

17 **extra_body support for OpenAI-compatible providers** NEW

50

A new `extra_body` parameter lets OpenAI-compatible providers receive provider-specific request parameters.

— Names parameter but no usage example. sdk-typescript-v0.1.25

18 **Idle CSI Workspace reclaim and reuse** NEW

45

Administrators can reclaim and reuse idle resident CSI Workspaces, with cleanup verification and atomic release.

— Describes mechanism but no admin command named. v0.26.3

19 **Zed editor extension for agent server** NEW

40

A new Zed extension connects to the Qwen Code agent server from the Zed editor.

— Names integration but no install instructions. sdk-typescript-v0.1.25

20 **Turn start and completion events in mods** NEW

35

Mods can now observe turn start and completion events, enabling lifecycle-aware extensions.

— Names event hooks but no API signature. desktop-v0.26.3

21 **Concurrent runner for batch CLI execution** NEW

35

A new concurrent runner executes batch CLI tasks in parallel.

— No flag or command name given. sdk-typescript-v0.1.25

22 **Experimental LSP support for code intelligence** NEW

30

Experimental LSP support enables code intelligence within sessions.

— Thin, no setup detail given. sdk-typescript-v0.1.25

23 **Durable terminal identity lookup in core** NEW

30

Core now performs durable terminal identity lookup for stable agent identification across sessions.

— Thin, no mechanism beyond the name. desktop-v0.26.3

24 **Opt-in memory extraction cadence after no-op run** NEW

30

A new opt-in extraction cadence feature controls memory extraction cadence after a no-op run.

— Thin, no config key or control named. `v0.26.3`

---

25 **Pinned extension package management for managed agents** NEW

25

Managed agents now support pinned extension package management.

— Bare description, no mechanism or command. `desktop-v0.26.3`

└──▷ **BREAKING ON UPGRADE**

! Settings keys previously named with `disable*` prefixes are renamed to `enable*` (positive naming convention); any configuration relying on the old `disable*` key names will break.

! The `read_many_files` tool is removed; its functionality moves to a `readManyFiles` utility available only via user `@-commands`.

WAS THIS USEFUL?  

Zed

Feedback ⓘ v1.26.2 NOTES ↗ RANK

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

> Zed v1.26.2 adds Pyrefly as a built-in Python language server, lets the project panel sort files by extension, enables Display P3 wide-gamut color rendering on macOS, and extends the JavaScript debugger with DWARF support for WebAssembly plus a richer variable-inspection hover.

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

01 Pyrefly as built-in Python language server NEW 80

Zed can now use Pyrefly as the Python language server by setting `language_servers` to `pyrefly` in the Python language configuration, e.g. `{"languages": {"Python": {"language_servers": ["pyrefly"]}}}`.

Switch your Python project to use Pyrefly instead of the default language server for faster type-checking.

json

```
{
  "languages": {
    "Python": {
      "language_servers": ["pyrefly"]
    }
  }
}
```

— Names exact config key and value with runnable snippet v1.26.2

02 Project panel sort by file extension NEW 75

The project panel can now group files by extension using the new `extension` option in `project_panel.sort_order`, configured as `{"project_panel": {"sort_order": "extension"}}`.

Group project panel files by extension so all .rs, .toml, and .md files cluster together.

json

```
{
  "project_panel": {
    "sort_order": "extension"
  }
}
```

— Exact config key and value with runnable snippet v1.26.2

03 **Display P3 wide-gamut color support on macOS** NEW 70

Zed adds support for the Display P3 wide-gamut color space on macOS, enabled via the `use_display_p3` setting (`{"use_display_p3": true}`) for more accurate color rendering on compatible displays.

Use the full Display P3 color space on a wide-gamut macOS display for more accurate colors in the editor.

```json
{
  "use_display_p3": true
}
```

— Named setting with runnable config snippet v1.26.2

THINNER COVERAGE BELOW

04 **DWARF debugging for WebAssembly** NEW 45

Zed adds DWARF debugging support to the JavaScript debugger, enabling debugging of WebAssembly sources.

— Names mechanism but no config or command shown v1.26.2

05 **Expandable variable tree in debugger hovers** IMPROVED 35

Debugger hovers now show an expandable, keyboard-accessible variable tree for inspecting nested values.

— Describes behavior but no named surface or steps v1.26.2

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ

announced · Oct 27, 2026 BLOG ↗ RANK

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

> Graphify launched a Perplexity Computer skill that lets users map any public GitHub repository into a code knowledge graph directly from a chat session.

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

01 Perplexity Computer skill for GitHub repo graphs NEW

85

A new `/graphify` command in Perplexity Computer maps any public GitHub repository into a code knowledge graph of files, functions, classes, calls, and imports, with no local install required. The skill is installed from the Perplexity Computer skills marketplace under Plugins > Skills to enable `/graphify` for any Computer session.

Map an unfamiliar public repository so Perplexity Computer can answer architecture questions grounded in the actual call graph.

```
$ /graphify https://github.com/<owner>/<repo>
```

— Names exact command, install path, and graph scope; mechanism beyond that is unspecified.

launch-20261027-2db5723a

WAS THIS USEFUL? 👍 👎

Replit Agent

Feedback ⓘ announced · Oct 27, 2026 BLOG ↗ RANK

Replit Agent is an AI coding agent that builds, tests, and deploys applications in Replit.

> Replit Agent's biggest release this window opens Cloud Agents up to entire company repositories, letting non-engineers propose tested, automatically-reviewed pull requests from Slack, mobile, or a web connector, alongside a new self-improving `skills` memory system and in-sandbox browser testing.

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

01 Cloud Agents for existing company repositories NEW

88

Connect an existing repository at `replit.com/repos` to let non-engineers describe changes in plain language; Cloud Agents explore the codebase, configure an initial VM with repos, runtimes, dependencies, and credentials pulled from external secrets managers via a credential broker, and produce a tested pull request in GitHub or GitLab. Available to all Pro & Enterprise customers, and every contributor starts from that same running, current app without individual setup.

When onboarding a new team or repository so all contributors share one running environment without individual setup.

📍 Visit replit.com/repos, connect your repository, and let Cloud Agents explore the codebase and configure the initial VM. All contributors then start from that shared, automatically updated environment.

— Names endpoint, VM setup mechanism, credential broker, access tier

launch-20261027-2107699a

02 Agent interactions and pull requests via Slack NEW

78

Mention or DM `@Replit` in Slack to start an investigation or open a pull request from a conversation thread, with progress and feedback continuing in the same thread. Set up by going to Workspace Settings, linking your Replit account, and choosing a workspace.

When a support lead or designer wants to propose a fix without setting up a local environment, go to Workspace Settings to link Replit to Slack, then DM or mention the bot to open a pull request from the conversation.

📍 In Replit, go to Workspace Settings › link your Replit account › choose a workspace. Then in Slack, DM or mention @Replit with a description of the change. Follow progress in the thread; the agent opens a tested pull request in GitHub or GitLab with screenshots and video attached.

— Exact command and setup path given launch-20261027-2107699a

03 **Self-improving workspace memory (skills), in early access**

NEW

65

In early access: when engineers approve review feedback, agents draft or update written instructions (skills) that all future Cloud Agent contributions follow. Replit reports this reduced token usage by 51% and cut a common data task from 16 minutes to 1 minute internally.

— Concrete metrics but no enablement steps given launch-20261027-2107699a

THINNER COVERAGE BELOW

04 **Automated in-sandbox testing with screenshots and video** NEW

50

Cloud Agents now test each change in isolated cloud sandboxes by opening the running app in a browser, clicking through the change, and attaching screenshots and video to the pull request before it reaches a reviewer.

— Explains mechanism but no named config or command launch-20261027-2107699a

05 **Plain-language codebase Q&A for non-engineers** NEW

35

Replit now lets anyone query the company codebase in plain language to get answers grounded in what is actually built, accessible to support, marketing, and sales roles.

— No command, endpoint, or UI path specified launch-20261027-2107699a

06 **Start pull requests from the Replit mobile app** NEW

30

Start a pull request from the Replit mobile app when an idea comes up away from a desk.

— Bare feature name, no mechanism or steps launch-20261027-2107699a

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ announced · Oct 26, 2026 BLOG ↗ RANK

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

> Kiro completes its Claude 5.5 model family by adding Claude Haiku 5.5 across IDE, CLI, Web, and Crew for fast, high-volume tasks.

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

01 **Claude Haiku 5.5 model support** NEW 78

Kiro now supports Claude Haiku 5.5 across IDE, CLI, Web, and Crew for high-volume, speed-sensitive tasks such as context compaction, bulk text processing, classification, and small edits across many files. It runs with Adaptive Thinking across five effort levels (low to max, defaulting to medium) and a 1M-token context window, with requests over 100K tokens billed at a 0.3x credit multiplier. It is available to Kiro Pro, Pro+, Pro Max, and Power customers in `us-east-1` (N. Virginia) and `eu-central-1` (Frankfurt) at a 0.06x credit multiplier.

— Names model, effort levels, context window, regions and pricing multipliers.

launch-20261026-ebd3f272

WAS THIS USEFUL? 👍 👎

Anysphere Cursor

Feedback ⓘ announced · Oct 20, 2026 BLOG ↗

RANK

Cursor is a built to make you extraordinarily productive, agents turn ideas into code. Accelerate development by handing off tasks to Cursor.

> Cursor launched the Agent SDK, a TypeScript framework for defining, testing, and deploying custom agents to Cursor infrastructure or self-hosted environments, with built-in support for PR-triggered workflows and multi-agent orchestration.

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

01 Define and deploy custom agents with Agent SDK NEW

95

Agents are defined in TypeScript using `defineAgent` from `@cursor/agent-sdk`, with prompts written in Markdown and tools implemented as TypeScript functions. The model is configured via the `model` key, including `id`, `params` entries such as `effort` and `fast`, alongside a `local.cwd` working directory. Running `agent-sdk deploy` builds the agent, gives it a URL, and starts it on Cursor infrastructure.

Define and configure a new agent locally before deploying — use `defineAgent` to set the model, effort, and working directory.

typescript

```typescript
import { homedir } from 'node:os';
import { join } from 'node:path';
import { defineAgent } from '@cursor/agent-sdk';

export default defineAgent({
  name: 'Robo',
  model: {
    id: 'composer-2.5',
    params: [
      { id: 'effort', value: 'high' },
      { id: 'fast', value: 'true' },
    ],
  },
  local: {
    cwd: join(homedir(), '.cache', 'factory', 'robo'),
  },
});
```

After iterating locally and validating with evals, deploy the agent to Cursor infrastructure to give it a live URL.

```
$ agent-sdk deploy
```

— Names exact command and config keys with runnable example launch-20261020-3303f22e

THINNER COVERAGE BELOW

02 **Trigger agents from pull request events** NEW 55

Agents can be triggered from pull request events on GitHub, GitLab, or Bitbucket to automate code review, risk assessment, and routing workflows.

— Names three platforms but no config surface shown launch-20261020-3303f22e

03 **Self-hosted agent deployment** NEW 50

Agents can run on your own infrastructure as a single Node process that can be packaged in a container, independent of Cursor-hosted deployment.

— Names Node process and container but no exact command launch-20261020-3303f22e

04 **Multi-agent subagent orchestration** NEW 40

The SDK supports multi-agent patterns: handing tasks off to specialized subagents with their own tools and sessions, or dividing work among subagents running in parallel.

— Mechanism described but no named API or flag launch-20261020-3303f22e

05 **Evals and event replay for agent development** NEW 35

Agents support running evals and event replay out of the box for iterative development and workflow validation before deploying.

— No command or config named, prose only launch-20261020-3303f22e

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Arcade

Feedback ⓘ snapshot-20261029 NOTES ↗ RANK

Arcade provides an actions runtime that lets AI agents securely call external services with authorization and governance controls.

> Arcade shipped Markdown drafting and search upgrades across its Slack and Mail MCP Apps, tightened SSRF-style hostname blocking for Gmail and Drive, and introduced two breaking changes around Outlook body formatting and Slack OAuth scopes.

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

01 **Outlook Mail defaults to Markdown-as-HTML bodies** BREAKING

80

Outlook Mail now defaults to sending Markdown rendered as HTML instead of plain text; pass `body_format=plain_text` to retain the previous plain-text behavior.

— Clear before/after with an exact flag to restore old behavior. snapshot-20261029

02 **Mail draft App enhancements for Gmail and Outlook** NEW

70

Gmail and Outlook Mail draft Apps now support autosave, Markdown lists, and moving recipients between To, Cc, and Bcc fields; Outlook Mail also adds a `CreateForwardDraft` tool to open reply and forward drafts directly in the draft App.

— Groups thin mail-draft additions while preserving the named `CreateForwardDraft` tool.

snapshot-20261029

03 **DNS-rebind and private-IP hostname blocking in Gmail and Drive**

NEW

60

Gmail and Google Drive now refuse known DNS-rebind and private-IP-embedding hostnames before resolving them, independent of the engine's outbound guard.

— Explains mechanism and scope but no configurable surface to act on. snapshot-20261029

THINNER COVERAGE BELOW

04 **Slack search channel and visibility filters** NEW

55

Slack search adds an `only_my_channels` filter and a new public-only search tool.

— Named filter and tool are directly usable, but little explanation of behavior.

snapshot-20261029

05 Markdown draft composition in Slack via MCP App NEW

45

Slack now supports composing, editing, and scheduling Markdown messages using a new draft card within the MCP App.

— Names the draft card mechanism but no exact command or flag. snapshot-20261029

06 Slack search tools require new OAuth scopes BREAKING

40

Slack search tools now require new OAuth scopes, so existing connected accounts must be re-authorized to continue using search.

— States the re-authorization action but no scope names or steps. snapshot-20261029

└──▷ BREAKING ON UPGRADE

! Outlook Mail now defaults to Markdown-as-HTML bodies instead of plain text; pass `body_format=plain_text` to keep the previous behavior.

! Slack search tools require new OAuth scopes — existing accounts must be re-authorized.

WAS THIS USEFUL? 👍 👎

CrewAI

Feedback ⓘ 1.15.33 NOTES ↗ RANK

Open-source Python framework for building multi-agent workflows using role-based AI agent Crews and event-driven Flows.

> CrewAI's CLI gains three new capabilities: discovering organization skills, passing runtime inputs to flows, and zip-based deployment with commit tracking.

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

01 **Zip deployment with commit reporting** NEW 70

Build and deploy a zip artifact with `deploy create --zip`, which packages the current working tree and also prints the commit hash the zip was built from for audit trail purposes.

Package the current working tree as a zip and deploy it, capturing the commit it was built from for audit trail.

```
$ crewai deploy create --zip
```

— Exact flag, behavior, and purpose named; no further scope details 1.15.33

02 **Skill discovery via `crewai skill search`** NEW 65

List your organization's available skills with the new `crewai skill search` command, letting you browse registered skills before wiring one into a crew.

Browse skills your organization has registered before wiring one into a crew.

```
$ crewai skill search
```

— Names exact command; mechanism of skill registration unexplained 1.15.33

03 **Runtime inputs for flows via `--inputs`** NEW 65

Pass runtime inputs to a Python flow using `crewai run --inputs`, useful for supplying parameters like targets without hardcoding them, e.g. in CI pipelines.

Supply runtime parameters to a Python flow without hardcoding them — useful in CI pipelines.

```
$ crewai run --inputs '{"target": "example.com", "depth": 3}'
```

— Exact flag and example shown, but underlying flow mechanism not detailed 1.15.33

WAS THIS USEFUL? 👍 👎

Daily Pipecat

Feedback ⓘ v1.15.0 NOTES ↗

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 adds expressive conversation (alpha), classifier-based turn detection, audio quality scoring, predictive backend delegation, and AssemblyAI TTS.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.15.0 https://github.com/pipecat-ai/pipecat.git
# already have the repo? check out this version:
$ git checkout v1.15.0
```

UPDATED INTEGRATION Enable expressive conversation (alpha) with PipelineParams(enable_expressive_conversation=True): TTS services perform expression tags the LLM writes — `<<excited>>` sets delivery, `[[laugh]]` inserts a sound, `[[pause 500]]` pauses in milliseconds, and `^^word^^` stresses a word — instead of speaking them aloud.

Control which expression tags an LLM offers via PipelineParams(expression_policy=...) with an `ExpressionPolicy`: its `delivery` and `sounds` fields list which tags are offered by universal name, and its `guidance` lines (presets such as `ExpressionGuidance.MATCH_STYLE` and `ExpressionGuidance.SLOW_FOR_DETAILS`, or custom strings) say when to use them.

Add `PipecatToneReader` after the STT service to read each transcribed user turn with the Pipecat Tone model, setting emotions and sarcasm likelihood on `TranscriptionFrame`. Install with `pipecat-ai[tone]`.

Add perception to expressive conversation (alpha): STT services or downstream processors set a `perception` field of `PerceptionSignal`s on `TranscriptionFrame`, `InterimTranscriptionFrame`, and `EagerTranscriptionFrame`; the user aggregator writes the turn's final reading into `UserTurnStoppedMessage.perception`.

Use `ClassifierUserTurnStopStrategy` and `ClassifierUserTurnStrategies` to let a classifier such as `JevClassifier` decide whether a user turn is complete, as an alternative to `FilterIncompleteUserTurnStrategies`.

+3 more

NEW INTEGRATION AssemblyAI Streaming TTS integration

NEW INTEGRATION ai-coustics Tyto audio quality scoring integration

UPDATED INTEGRATION AWS Bedrock LLM reasoning output support

UPDATED INTEGRATION NVIDIA TTS custom model configuration

UPDATED INTEGRATION Smallest TTS number pronunciation language setting

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.214 NOTES ↗ RANK

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

> HyperFrames v0.8.214 adds lane-wide colour grading via grade groups and the ability to export colour grades as 3D LUTs.

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

01 **Lane-wide colour grading with grade groups** NEW 67

Apply one colour grade to an entire lane of clips using `hf-color-group`, so editing one element re-grades every clip in that lane.

– Names the config key and explains the mechanism clearly. v0.8.214

THINNER COVERAGE BELOW

02 **3D LUT export to .cube files** NEW 55

Bake a colour grade into a 3D LUT and export it as a `.cube` file that matches the preview.

– Names format and behaviour but no exact command shown. v0.8.214

WAS THIS USEFUL? 👍 👎

Mastra

Feedback ⓘ @mastra/core@1.80.0 NOTES ↗ RANK

Mastra is a TypeScript framework for AI applications and agents, providing workflows, agent memory, RAG, tool calling, and evals.

> Mastra's core package adds a span-aggregation endpoint and richer observability fields, model-provider error classification, render-only tools that stream UI without a client response, and agent experiments that swap models without touching the agent definition — alongside four breaking API changes in tracing, connect, code-sdk, and the playground UI.

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

01 Span aggregation and observability fields NEW 90

buildSpanPreviews() from `@mastra/core/observability` generates input, output, and error previews from a processed span, gated by the `observability-span-previews` core feature flag. Span records gain nine new native columns: `externalParentSpanId`, `resumedFromSpanId`, `model`, `provider`, `outcome`, `stopReason`, `stopReasonContext`, `timeToFirstTokenMs`, `errorCount`. A new `POST /observability/spans/aggregate` endpoint plus aggregateSpans() aggregate span metrics (counts, latency, error rate, tokens, cost) in ClickHouse, Postgres, and DuckDB stores.

— Endpoint, function and all nine field names enumerated verbatim. @mastra/core@1.80.0

02 Agent experiments with model override NEW 85

Passing `agentModel` to dataset.startExperiment() or dataset.startExperimentAsync() runs an experiment against a different model without editing or versioning the agent; results are recorded in `metadata.agentModel` and each run's trace metadata.

Test how a support agent performs with a cheaper model without changing or versioning the agent definition.

typescript

```typescript
await dataset.startExperiment({
  targetType: 'agent',
  targetId: 'support-agent',
  agentModel: 'anthropic/claude-haiku-5-5',
  scorers: ['accuracy'],
});
```

— Exact API and runnable example given, mechanism explained. @mastra/core@1.80.0

03 Provider error classification utilities NEW 80

`classifyProviderError`, `providerErrorReason`, and `toPersistedErrorIdentity` from `@mastra/core/error` classify model-provider failures (authentication, quota, rate limit, context overflow, timeout, outage); `providerRequestId` and `providerModelId` extract the provider's request ID and model name from errors.

Branch retry logic on the specific reason a model-provider call failed, using the new classification API.

```typescript
import { classifyProviderError } from '@mastra/core/error';

switch (classifyProviderError(error)) {
  case 'rate_limited':
    // wait, then retry
    break;
  case 'quota':
    // ask for more credit
    break;
  default:
    // show the provider's reason
}
```

— Names all functions and failure categories with a code example. `@mastra/core@1.80.0`

04 **Channel adapter handling and approval controls** NEW

75

A new `onMessage` handler in `ChannelHandlers` handles messages that do not mention the bot (calling `defaultHandler` runs the agent and subscribes the thread); a `textDisplay: 'hidden'` adapter setting suppresses the agent's reply text while tool display, approval prompts, errors, and typing status still post; and new SlackProvider({ toolApprovers: 'anyone' }) lets any channel member, not just the requester, approve or deny pending tools in Slack.

— Three named config surfaces fully enumerated, no code example. `@mastra/core@1.80.0`

05 **Render-only tools in createTool** NEW

65

Setting `renderOnly: true` in createTool() creates tools that stream display calls to the browser without requiring a client result.

Stream display calls for a product card to the browser while the agent keeps running, without waiting for a client response.

```typescript
import { createTool } from '@mastra/core/tools';
import { z } from 'zod';

const showCard = createTool({
  id: 'show-card',
  description: 'Display a product card',
  inputSchema: z.object({ productId: z.string() }),
  renderOnly: true,
});
```

— Flag named with a runnable example, but behaviour scope is narrow. `@mastra/core@1.80.0`

06 **code-sdk no longer auto-registers built-in subagents** BREAKING

60

`@mastra/code-sdk` no longer registers built-in subagents by default; pass `subagents: [...builtinSubagents]` explicitly to restore prior behavior.

— Exact migration fix provided alongside the behavior change. `@mastra/core@1.80.0`

THINNER COVERAGE BELOW

07 **LLM request processor provider ID and merging control** NEW

55

`processLLMRequest` processors now receive the `apiProviderId` of the model actually called (e.g. `groq.chat` or `openai.responses`) alongside the existing `model.provider` field; `errorProcessorDefaults: false` disables system-message merging for OpenAI-compatible Chat Completions providers.

— Named fields and flag, but no usage example. `@mastra/core@1.80.0`

08 **queryTraceThreads field removal and pagination cap** BREAKING

55

queryTraceThreads() thread rows no longer include `model` or `provider`, and `page.limit` is now capped at 100 (default 25); limits above 100 fail validation.

— States exact before/after and new default/cap values. `@mastra/core@1.80.0`

09 **Thread-run-status subscription chunks** NEW

50

Setting `withRunStatus: true` on thread subscriptions receives `thread-run-status` chunks (`success`, `failed`, `aborted`, or `suspended`) without closing the subscription.

— Flag and status values named but no example. `@mastra/core@1.80.0`

10 **AgentController default mode lookup** NEW 50

AgentController.getDefaultMode() reads the mode new agent sessions start in without creating a session, following `defaultModeId`, a mode marked as default, then the first mode.

— Method and fallback order named, no example given. `@mastra/core@1.80.0`

11 **Connect sessions endpoint response field renamed** BREAKING 50

`POST /connect/:provider/sessions` in `@mastra/connect` now returns `launchUrl` instead of `connectUrl`.

— Endpoint and field names given, no migration guidance. `@mastra/core@1.80.0`

12 **MIME-type lookup utility** NEW 45

`getMimeType`, importable from `@mastra/core/workspace`, returns the extension-based MIME type (e.g. `'image/png'`) or `'application/octet-stream'` as a fallback, now usable outside the filesystem provider.

— Named function but thin description of behaviour. `@mastra/core@1.80.0`

13 **Read sandbox process output without dismissal** NEW 45

Calling get(pid, { release: false }) on the sandbox process manager reads a finished sandbox process's output without dismissing it.

— Exact call signature given but no further mechanism detail. `@mastra/core@1.80.0`

14 **TracesExplorer sorting API consolidated** BREAKING 40

`@mastra/playground-ui` `TracesExplorer` sorting API now takes a single `{ field, direction }` object, replacing the previous `sortDirection` and `onSortDirectionChange` props.

— Names old and new props but no code example. `@mastra/core@1.80.0`

└──▷ BREAKING ON UPGRADE

! queryTraceThreads() thread rows no longer include `model` or `provider`, and `page.limit` is now capped at 100 (default 25); limits above 100 fail validation.

! `POST /connect/:provider/sessions` in `@mastra/connect` now returns `launchUrl` instead of `connectUrl`.

! `@mastra/code-sdk` no longer registers built-in subagents by default; pass `subagents: [...builtinSubagents]` explicitly to restore prior behavior.

! `@mastra/playground-ui` `TracesExplorer` sorting API now takes a single `{ field, direction }` object, replacing the previous `sortDirection` and `onSortDirectionChange`

props.

WAS THIS USEFUL? 👍 👎

Microsoft Semantic Kernel

Feedback ⓘ python-1.46.0

NOTES ↗ RANK

Semantic Kernel provides an open-source SDK for building AI agents and integrating language models into C#, Python, and Java applications.

> Semantic Kernel's Python SDK (1.46.0) improves OpenAPI plugin parsing, adds typing.Literal schema generation, expands PostgreSQL vector store filtering, and fixes tool-choice exclusion across several model providers.

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

01 typing.Literal support in KernelJsonSchemaBuilder NEW

70

`KernelJsonSchemaBuilder` now supports `typing.Literal` so functions annotated with literal types generate accurate JSON schemas for tool definitions, restricting tool arguments to a fixed set of values.

Annotate a kernel function parameter with `typing.Literal` so the generated JSON schema restricts the tool argument to a fixed set of values.

python

```
from typing import Annotated, Literal
from semantic_kernel.functions import kernel_function

class SearchPlugin:
    @kernel_function(description="Search with a specific mode.")
    def search(
        self,
        mode: Annotated[Literal["fast", "deep", "hybrid"], "The
search mode to use."]
    ) -> str:
        return f"Running {mode} search"
```

— Named class and a runnable code example provided. python-1.46.0

02 include_in_function_choices honored across model providers

IMPROVED

60

The `include_in_function_choices` flag is now honored by the Anthropic, Bedrock, Google AI, and Vertex AI tool formatters, so functions marked to be excluded are correctly omitted from tool lists sent to those providers.

— Names the flag and four providers but no example shown. python-1.46.0

THINNER COVERAGE BELOW

03 **Callable and mixed filters in PostgreSQL vector store**

IMPROVED

55

The PostgreSQL vector store now accepts callable and mixed search filters, expanding the range of filter expressions usable in `get_nearest_matches` queries.

— Names the exact method but gives no usage example. python-1.46.0

04 **OpenAPI path-item parameter inheritance** IMPROVED

50

The OpenAPI parser now supports path-item parameters and non-method keys, letting plugins built from OpenAPI specs correctly inherit shared parameters defined at the path level rather than only on individual operations.

— Describes mechanism but no code example or config key shown. python-1.46.0

WAS THIS USEFUL? 👍 👎

nanobot

Feedback ⓘ v0.4.0 NOTES ↗

Nanobot is a self-hosted personal AI assistant framework with tools, memory, scheduled tasks, and integrations with chat applications.

nanobot v0.4.0 adds SQLite conversation history (imported from JSONL), an in-app update flow with confirmation, Mermaid diagram viewing in the TUI, remote workbench connections, local file and video attachments, and `nanobot gateway stop` with `--no-dock`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.4.0 https://github.com/HKUDS/nanobot.git
# already have the repo? check out this version:
$ git checkout v0.4.0
```

The gateway now stores model history and browser transcripts in SQLite, importing existing JSONL history with source backups retained.

Stop the gateway cleanly with `nanobot gateway stop`; suppress the macOS Dock icon with `--no-dock`.

View, zoom, pan and copy Mermaid diagrams in the terminal TUI without leaving a conversation.

Connect the browser workbench to remote instances with explicit host identity, pairing and reconnect feedback.

Attach local files and video in the browser workbench, with image thumbnails, file cards and Markdown source views.

+3 more

NEW INTEGRATION Linear agent channel

NEW INTEGRATION WhatsApp Agent Platform channel

UPDATED INTEGRATION Discord replyToMessage parity with Telegram

NEW SUPPORTED PLATFORM Native platform wheels for macOS, Linux and Windows

└──▷ BREAKING ON UPGRADE

! Older versions cannot read new SQLite display history; a downgrade requires a stopped-writer backup-restoration procedure.

! The `apiType` / `api_type` OpenAI selector is converted to `api`; save the current format during the v0.4.0 window.

! The `write_stdin` compatibility shim has been removed.

! JSONL-only installations must pass through an import-capable 0.4.x release before 0.5.0.

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

Model Context Protocol Servers

Feedback ⓘ v2.2.0

NOTES ↗ RANK

MCP Servers provides reference server implementations for connecting AI applications to external tools and data.

> MCP Servers v2.2.0 adds a flag to run mcp-server-fetch without Node.js, and extends server-filesystem's file info tool with hard-link counts.

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

01 **Pure-Python mode for mcp-server-fetch** NEW 75

The new `--no-readability-js` flag disables Node.js-based HTML extraction in `mcp-server-fetch`, letting it run as a pure-Python server, e.g. `uvx mcp-server-fetch --no-readability-js`.

Run mcp-server-fetch without Node.js installed by disabling the readability extraction path entirely.

```
$ uvx mcp-server-fetch --no-readability-js
```

— Exact flag and runnable command given. v2.2.0

THINNER COVERAGE BELOW

02 **Hard-link counts in get_file_info** IMPROVED 45

The `get_file_info` tool in `@modelcontextprotocol/server-filesystem` now reports the hard-link count for each file.

— Names the tool and field but no usage example. v2.2.0

WAS THIS USEFUL? 👍 👎

gram

Feedback ⓘ server@2.29.0 NOTES ↗ RANK

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

> Gram's two releases added tool-call approval workflows and a risk-findings analytics dataset to Guardrails, an experimental Code Mode for MCP gateways, a unified MCP server management page, flexible gateway tool-serving modes, and admin-side upstream OAuth enforcement plus remote-server promotion controls.

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

01 Unified MCP server management page NEW

80

Hosted MCP servers are now managed from a single page with Tools, Resources and Prompts tabs, per-tool Team Access, hosted Settings sections, and a credentials list showing each upstream credential's kind, on/off status, which other servers share them, and Can use / Blocked rules for per-credential access control.

Update an existing API key's access policy to restrict it to specific auth clients and projects in your organization.

```
$ POST /keys.updatePolicy
```

— Enumerates every tab, section and access-control rule on the page dashboard@0.143.0

02 Upstream OAuth enforcement and migration reporting

NEW INTEGRATION NEW

80

Staff can turn on upstream auth enforcement per organization from the upstream OAuth migration page once auth clients, credential choices and assistant profiles are recorded; enforcement serves each request with the typed auth client its credential chose, and a credential with no usable choice is asked to reconnect rather than served with another client. Assistant turns bind a versioned execution-selection profile recording which typed auth client each MCP server's turns use, kept separately for identified-user and workload turns, and the migration page now names the single typed auth client serving each live credential's server choice while reporting coverage by credential class: interactive, ID-JAG, dashboard-minted, workload, and agent sessions.

— Deep mechanism and named credential classes, but staff-only toggle admin@0.10.0

03 Risk findings dataset and dashboard in Explore NEW

75

A new `risk_findings` dataset in Explore lets organization administrators count what risk policies caught, broken down by category, rule, detector, severity, policy, agent, team and person; a built-in Security & risk dashboard surfaces findings, high and critical findings, people affected, findings over time, top rules, and findings by agent, person and policy.

— Fully enumerates dataset dimensions and dashboard widgets `dashboard@0.143.0`

04 **Tool-serving modes for gateway connections** NEW 70

When approving a gateway connection you can choose All tools (live), All tools (frozen), or Specific tools, with frozen and specific tools served from a fast cache for faster tool access.

— Names the three modes and caching behaviour, clear choice point `dashboard@0.143.0`

05 **Staff override for remote MCP server promotion** NEW 70

Staff can explicitly promote remote MCP servers despite health eligibility criteria through the admin dashboard and Admin MCP, with the override recorded in the audit log; automatic eligibility reviews now run weekly while setup-docs delivery remains hourly.

— Names surfaces and review cadence, staff-only action `admin@0.10.0`

06 **Retired OAuth proxy link no longer honored** BREAKING 70

The server no longer reads the retired OAuth proxy link on toolsets — hosted MCP servers carrying that link are no longer reported as requiring OAuth unless they have a sign-in provider or an external OAuth server, and `oauth_proxy` is no longer reported as legacy authentication in admin server health.

— Names the retired field and the exact health-reporting change `admin@0.10.0`

07 **Code Mode discovery for MCP gateways** NEW 65

Code Mode runs ephemeral Python with tool search, schema descriptions and live authorized calls against an MCP gateway. It is off by default and rolled out per organization by Speakeasy staff from the admin dashboard's organization Features page.

— Mechanism described but enablement is staff-only, not self-service `dashboard@0.143.0` `admin@0.10.0`

08 **Require-approval policy for tool calls** NEW 60

Guardrails adds a Require approval for tool calls policy that routes held calls to approvers via a Tool approvals tab and Slack DMs, gated behind the `risk-tool-approvals` flag.

Verify ahead of time whether your account can write policy grants for an agent, without actually creating anything.

```
$ GET /agents.checkPolicyGrants
```

— Names the flag and approval surfaces but not full approver logic `dashboard@0.143.0`

09 **Adoption Dashboard Link on org overview** NEW 60

Staff can store an Adoption Dashboard Link (must be `https`) on an organization's overview in the admin dashboard; the link is cleared by saving an empty value and is returned by the organization record and the staff Admin MCP organization summary.

— Names the https constraint and two surfaces that return it `admin@0.10.0`

THINNER COVERAGE BELOW

10 **Duplicate remote session clients** NEW 55

A remote session client can be duplicated from the Clients list or detail page, keeping original settings and optionally registering a new client with the authorization server when dynamic registration is supported.

— Describes behaviour and UI entry points, no API named `dashboard@0.143.0`

11 **Catalogue promotion length validation** IMPROVED 55

Catalogue promotion titles and descriptions are now validated against the 100-character catalogue limit, returning field-specific length errors through the admin API and Admin MCP.

— Names the limit and two surfaces, no endpoint path given `admin@0.10.0`

12 **Browse Admin MCP tools before connecting** NEW 55

Enabled staff Admin MCP tools can now be browsed and searched on the setup page before connecting a client, with descriptions and access labels for reads, approval flows, and direct writes.

— Names access-label categories and the setup page location `admin@0.10.0`

13 **MCP gateways available to all organizations** IMPROVED 50

MCP gateways no longer require the `gram-gateway-endpoints` rollout flag and are now available to every organization by default.

— Names the removed flag but little else `dashboard@0.143.0`

14 **Filter tool logs by agent** IMPROVED 45

Tool logs can now be filtered by `agent_ids` to narrow calls made by a specific agent acting on a person's behalf.

— Clear param but thin on behaviour beyond filtering `dashboard@0.143.0`

└──▷ **BREAKING ON UPGRADE**

! The server no longer reads the retired OAuth proxy link on toolsets — hosted MCP servers carrying that link are no longer reported as requiring OAuth unless they have a sign-in provider or an external OAuth server, and `oauth_proxy` is no longer reported as legacy authentication in admin server health.

`server@2.29.0` **NOTES ↗** **LATEST** **STABLE**

gram adds multi-endpoint hosted MCP servers, Code Mode for gateways, per-project approvals servers, auth client management, and several new API endpoints.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch server@2.29.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout server@2.29.0
```

Add multiple endpoints to a hosted MCP server with `mcpEndpoints.create` and remove them with `mcpEndpoints.delete`; an enabled server cannot lose its last serving endpoint, so disable it first.

Detach a user session issuer from any server type by setting `clear_user_session_issuer` on `mcpServers.update`; this cannot be combined with `user_session_issuer_id` and is refused while identity-chaining bindings depend on the issuer.

Query tool usage by hosted server with the new `mcp_server_ids` filter; results now include `mcp_server_id` on hosted server options, targets and traces. `hosted_toolset_slugs` still works but is now deprecated.

Check whether policy grants can be written without creating anything via the new `agents.checkPolicyGrants` endpoint, using the same rules as `agents.create` and `agents.createPolicyGrant`.

Update an API key's policy in place with the new `keys.updatePolicy` endpoint; the key's policy can name only auth clients and projects in its own organization, and every change is recorded in the audit log.

+16 more

└──▷ **BREAKING ON UPGRADE**

- ! `toolsets.update` now returns a bad request error if you pass `mcp_enabled`, `mcp_slug`, `mcp_is_public`, `network_access_mode` or `custom_domain_id`; move these calls to `mcpServers.update` or `mcpEndpoints.update`.
- ! `toolsets.checkMCPSlugAvailability` is removed.
- ! `toolsets.setUserSessionIssuer` is deprecated and will be removed on 2026-11-25; it now refuses to unlink an issuer that identity-chaining bindings depend on (409) and answers with `Deprecation` and `Sunset` headers.
- ! A hosted server's slug no longer follows its endpoint after an address change; look a hosted server up by its endpoint slug, not its server slug.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Providers & GPU Clouds

Crusoe Cloud

Feedback ⓘ snapshot-20261029 NOTES ↗ RANK

Crusoe Cloud provides GPU infrastructure and managed inference for AI training and deployment.

> Crusoe Cloud shipped bulk VM start/reset for up to 25 VMs at once and added CMK generation visibility across Kubernetes cluster and node pool CLI commands.

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

01 **Bulk VM start and reset** NEW 70

Crusoe Cloud now supports starting or resetting up to 25 VMs in a single operation, letting operators restart a batch of running VMs in one shot during a maintenance window instead of looping through individual reset calls, e.g. `crusoe compute vms reset vm-001 vm-002 vm-003 vm-004 vm-005`.

Restart a batch of up to 25 running VMs in one shot during a maintenance window instead of looping through individual reset calls.

```
$ crusoe compute vms reset vm-001 vm-002 vm-003 vm-004 vm-005
```

— Runnable command given but limit's enforcement mechanism unexplained snapshot-20261029

02 **CMK generation visibility in Kubernetes CLI** NEW 65

View the CMK generation for clusters and node pools via `crusoe kubernetes clusters list`, `crusoe kubernetes clusters get`, `crusoe kubernetes nodepools list`, and `crusoe kubernetes nodepools get`.

— Exact commands named but no example invocation shown snapshot-20261029

WAS THIS USEFUL? 👍 👎

Baseten

Feedback ⓘ Per-replica metrics NOTES ↗ RANK

Baseten deploys and serves machine learning models as autoscaling inference endpoints, with a model API catalog and dedicated GPU deployments.

> Baseten added per-replica visibility into the Metrics tab, letting users spot outliers in CPU, GPU, and memory usage that aggregate graphs hide.

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

01 **Per-replica CPU, GPU, and memory metrics** NEW 55

Enabling Per replica in the Metrics tab shows CPU, GPU, and memory usage broken down by individual replica across every workspace, helping identify a misbehaving replica whose spike is masked by aggregate metrics.

Identify a misbehaving replica whose memory spike is masked by aggregate metrics.

📍 In the Baseten console, open the Metrics tab for your deployment, then toggle on Per replica to compare CPU, GPU, and memory usage across individual replicas.

— Names exact UI toggle and metrics but no limits or mechanism. Per-replica metrics

WAS THIS USEFUL? 👍 👎

Cerebras Inference

Feedback

Dedicated models and API guidance

NOTES ↗

Cerebras Inference provides hosted language-model inference through shared APIs and dedicated endpoints on Cerebras hardware.

API responses now expose image-token usage in `usage.prompt_tokens_details.image_tokens`, already counted within `usage.prompt_tokens`.

NEW INTEGRATION New Dedicated models: Kimi K3, GLM 5.3 Flash, Inkling, Reflection AI Beam, MiMo V2.6 Pro

NEW SUPPORTED PLATFORM Image inputs in tool messages for Gemma 4 31B, Kimi K2.7 Code, and Kimi K3 Dedicated endpoints

NEW SUPPORTED PLATFORM Reasoning controls for Kimi K3 and GLM 5.3 Flash

WAS THIS USEFUL?

◆ Local LLM Runtimes

Ollama

Feedback ⓘ v0.40.13 NOTES ↗ CODE ↗ RANK

Ollama downloads, serves, and chats with local language models through a CLI and REST API.

> Ollama extended platform support for Liquid AI's LFM3 model family and wired text-to-speech generation into its REST API, alongside a small CLI UX tweak.

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

01 **Text-to-speech via /v1/audio/speech endpoint** NEW 90

Ollama's `/v1/audio/speech` endpoint now works with GGUF models by routing requests to llama-server's `/tts` API chunk by chunk, enabling text-to-speech on the `lfm3-audio` and `lfm3-omni` models.

Generate speech from text on Linux or Windows using the LFM3 audio model via the REST API.

```
$ curl http://localhost:11434/v1/audio/speech -d '{"model":
"lfm3-audio", "input": "Hello, this is a text-to-speech test."}'
```

— Named endpoint, mechanism, and runnable curl example provided v0.40.13

THINNER COVERAGE BELOW

02 **LFM3 model support on Linux and Windows** NEW 55

Ollama now runs Liquid AI's `lfm3`, `lfm3-audio`, and `lfm3-omni` models on Linux and Windows, in addition to the existing Apple silicon support.

— Names models and platforms but no setup command given v0.40.13

03 **CLI no-subcommand behavior change** IMPROVED 40

Running `ollama` without a subcommand now prints command help and exits, instead of launching the previous welcome flow.

— Clear before/after but no flag or config detail v0.40.13

WAS THIS USEFUL? 👍 👎

llama.cpp

Feedback ⓘ v0.8.0 NOTES ↗ CODE ↗ RANK

Llama CPP is a C/C++ library and set of command-line tools that runs large language models locally.

> llama.cpp v0.8.0 adds real-time and text-to-speech server endpoints, a new streaming audio API for multimodal models, and broader Chat API and sampler improvements, alongside two breaking changes to the CLI and the audio generation state API.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 **Realtime and TTS endpoints in server** NEW 80

Adds a new `/v1/realtime` endpoint to the server implementing the OpenAI realtime protocol for conversation and transcription, with `response.done` sent after committed audio drains, and a new `POST /tts` endpoint for text-to-speech models.

— Endpoints and timing behaviour named, no usage example given v0.8.0

02 **mtmd audio generation API overhaul** BREAKING 80

Improves the mtmd audio generation API by adding `gen_audio` to `mtmd_caps`, new TTS types `SOPRANO` and `LFM3A`, and repetition penalty for sampled codes, and replaces raw `state_data` buffers with the new `mtmd_gen_state` stateful object. Callers using `state_data` directly must migrate to `mtmd_gen_state`.

— Multiple named surfaces and a clear migration requirement v0.8.0

03 **Proxy support and concurrent model downloads** IMPROVED 65

Now honors `HTTP_PROXY` and `HTTPS_PROXY` environment variables and allows concurrent downloads in common.

Route downloads through a corporate proxy when pulling models from Hugging Face.

```
$ HTTPS_PROXY=http://proxy.corp.example:3128 llama serve -hf ggml-org/Qwen3.5-0.8B-GGUF
```

— Named env vars plus a runnable example command v0.8.0

04 **Streaming audio input in mtmd** NEW 60

Adds a new experimental `mtmd_audio_stream` API in `mtmd` that turns PCM audio fed as it arrives into decoder-ready embedding rows, enabling streaming audio input, and

adds mtmd_helper_get_capabilities() to query the capabilities of an mmproj file from helper code.

— Mechanism explained, API names given but no call example v0.8.0

THINNER COVERAGE BELOW

05 **Required model flags on the CLI** BREAKING 55

`--model` or `-hf` is now required on the CLI; invocations that omitted both flags will fail.

— Clear breaking behaviour with exact flags named v0.8.0

06 **Speculative decoding improvements in server** IMPROVED 55

Adds support for `logprobs` together with speculative decoding in the server, persists the dynamic ngram cache across speculative decoding sessions, and adds probabilistic DFlash2 drafts.

— Three concrete mechanisms named, no usage example v0.8.0

07 **New Chat API capabilities** NEW 45

Exposes reasoning effort levels of the chat template as selectable via the API, and adds support for deferred tools and `tool_reference` content parts in the Chat API.

— No endpoint, field or parameter name given Names the content part but no usage shown v0.8.0

08 **DRY sampler sequence breaker cap** NEW 45

Adds new `LLAMA_DRY_MAX_SEQ_BREAKERS` constant (64) to the DRY sampler, capping distinct sequence breakers.

— Constant and value named, no usage context v0.8.0

09 **Configurable memory margin in server router** IMPROVED 40

Adds a configurable device memory margin for dynamic unloading of models in the server router.

— No config key or flag name given v0.8.0

10 **LFM3 multimodal family and new architectures** NEW 25

Adds support for the LFM3 multimodal model family and a batch of new model architectures.

— Headline mention only, no architecture names or specifics v0.8.0

└──▷ **BREAKING ON UPGRADE**

! `--model` or `-hf` is now required on the CLI; invocations that omitted both flags will fail.

! `mtmd_gen_state` replaces the raw `state_data` buffers of the audio generation API — callers using `state_data` directly must migrate to `mtmd_gen_state`.

WAS THIS USEFUL? 👍 👎

◆ **Agentic Infrastructure**

AWS Amazon Bedrock AgentCore

Feedback ⓘ

Runtime: Scope access to a session or runtime (2026-10-27) BLOG ↗ RANK

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

> Amazon Bedrock AgentCore Runtime now lets session data access be scoped per session on V2 agent runtimes.

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

01 **Session context keys scope data access in Runtime** NEW

55

AgentCore Runtime adds session context keys on V2 agent runtimes, allowing sessions that share a single execution role policy to each be restricted to only their own data.

— Names mechanism and scope but no config/API details or example

Runtime: Scope access to a session or r…

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

LanceDB

Feedback ⓘ v0.41.0 NOTES ↗ RANK

LanceDB is an open-source database that stores multimodal data and supports vector search.

> LanceDB v0.41.0 locks down embedding models and custom endpoints behind a new authorization API, expands blob handling with new query modes and tier controls, and ships a wide set of breaking changes across remote APIs, listings, and core type validation.

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

01 Authorization API for embedding models and endpoints

BREAKING

95

`SentenceTransformerEmbeddings.trust_remote_code` now defaults to `False`, so reopening tables that use remote-code embedding models or custom endpoints raises `UntrustedEmbeddingConfigError` unless explicitly approved. Call registry.allow_remote_code() and registry.allow_endpoint() at the process level to re-enable trusted models and endpoints before reopening such tables.

Re-enable a previously trusted embedding model after upgrading, so table operations that compute embeddings continue to work.

python

```python
from lancedb.embeddings import get_registry

registry = get_registry()
registry.allow_remote_code("your-trusted-model", revision="your-reviewed-commit")
registry.allow_endpoint("https://your-trusted-endpoint.example")
```

— Full mechanism, exact names, and runnable example. v0.41.0

02 Blob query modes and tier configuration

IMPROVED

85

LanceDB adds a `blob_mode` parameter to `to_arrow` and `to_list` in Python and a `blobMode` option in Node.js queries to return raw blob bytes; Node.js queries also accept `ArrayBuffer`, `Blob` / `File`, and `URL` as blob column inputs. Blob tier thresholds can now be set via lancedb.blob() from Python and Rust, and blob layouts are normalized with support for external byte ranges — existing blob SQL inserts using explicit adapter schemas should be reviewed.

Retrieve raw blob bytes alongside other columns when querying a table in Python.

```python
results = table.search().to_arrow(blob_mode="bytes")
```

— Names every blob surface and config entry point. v0.41.0

03 **Cascade option for drop_database** NEW 80

A `cascade` option added to `drop_database` removes all tables contained in the database in one call, avoiding `database not empty` errors: await db.drop_database("my_database", cascade=True).

Drop a database and all its tables in one call, avoiding `database not empty` errors.

```python
await db.drop_database("my_database", cascade=True)
```

— Named flag with runnable example. v0.41.0

04 **Remote API validation and signature changes** BREAKING 65

`RemoteDBConnection.create_table` parameter order has changed; unsupported remote connection and table options are now rejected with an error instead of silently ignored; and entity names for remote APIs are now validated centrally, so previously accepted names may now be rejected.

— Names the affected method and validation behavior. v0.41.0

05 **Lazy iterators for table, namespace and database listings** 60

BREAKING

Table and namespace listing APIs, plus `list_databases`, now return lazy iterators instead of lists — callers relying on list semantics such as indexing or len() must update.

— Names affected APIs and the exact behavior change. v0.41.0

THINNER COVERAGE BELOW

06 **Automatic refresh for computed columns and materialized views** 50

IMPROVED

Computed columns and materialized views can now be configured with automatic refresh when declared; materialized views are now always fully refreshed on refresh rather than partially.

— Explains behavior change but no config key named. v0.41.0

07 **bool(Expr) now raises instead of silently dropping predicates** 45

BREAKING

bool(Expr) now raises an error instead of silently dropping predicates, surfacing previously hidden query logic bugs.

— Names the exact surface and behavior change. v0.41.0

08 **Unenforced primary key on remote tables** NEW 40

Remote tables now support `set_unenforced_primary_key`.

— Names the method but no usage detail. v0.41.0

09 **large_binary type in Function column signatures** NEW 40

Python Function column signatures now accept `large_binary` as a type.

— Names the type but no example given. v0.41.0

10 **Storage engine renamed from LSM to RTE** BREAKING 40

LSM APIs were renamed to MemWAL and then again to RTE (realtime engine); callers using the old names must update.

— Names the rename but gives no migration path. v0.41.0

11 **Full-text search indexes on JSON fields** NEW 35

FTS indexes can now be built directly on JSON fields.

— States the capability without API detail. v0.41.0

12 **Mixed-type tuple fields rejected in pydantic models** BREAKING 35

Pydantic models with tuple fields containing mixed element types are now rejected instead of silently accepted.

— Names the validation rule but no example. v0.41.0

13 **Search effort control on vector search** NEW 25

Vector search queries now expose a search effort control parameter.

— No parameter name or usage shown. v0.41.0

14 **Synchronous table method signatures aligned** BREAKING 20

Synchronous table method signatures have been aligned across the API; callers passing positional arguments may be affected.

— No specifics on which methods changed. v0.41.0

└──▷ **BREAKING ON UPGRADE**

! `SentenceTransformerEmbeddings.trust_remote_code` now defaults to `False`; reopening tables with remote-code models or custom endpoints raises `UntrustedEmbeddingConfigError` unless registry.allow_remote_code() or registry.allow_endpoint() is called first.

! `list_databases` now returns a lazy iterator instead of a list.

! Table and namespace listing APIs now return lazy iterators.

! LSM APIs were renamed to MemWAL and then again to RTE (realtime engine) — callers using the old names must update.

! `RemoteDBConnection.create_table` parameter order has changed.

! Unsupported remote connection and table options are now rejected with an error.

! Entity names are now validated centrally for remote APIs — previously accepted names may now be rejected.

! Synchronous table method signatures have been aligned — callers passing positional arguments may be affected.

! bool(Expr) now raises instead of silently dropping predicates.

! Tuple fields with mixed element types in pydantic models are now rejected.

! Blob layouts are normalized and external byte ranges are now supported — existing blob SQL inserts with explicit adapter schemas must be reviewed.

! Materialized views are now always fully refreshed on refresh.

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.144.0 NOTES ↗ RANK

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

> Docling adds CLI control over heading hierarchy detection and introduces automatic OCR recovery for PDFs with broken text layers.

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

01 **Heading hierarchy detection via CLI flag** NEW 80

Docling can now run heading style signal detection without requiring `generate_parsed_pages`, and heading hierarchy can be controlled via the new `--heading-hierarchy` CLI flag, e.g. `docling --heading-hierarchy https://arxiv.org/pdf/2206.01062` to get structured outlines from documents lacking reliable PDF headings.

Convert a PDF with explicit heading hierarchy detection when you need structured outlines from documents that lack reliable PDF headings.

```
$ docling --heading-hierarchy https://arxiv.org/pdf/2206.01062
```

— Named flag with runnable example command. v2.144.0

THINNER COVERAGE BELOW

02 **Automatic OCR replacement for broken PDF text layers** IMPROVED 35

Docling now automatically replaces broken PDF text layers with OCR, recovering readable text from corrupted or malformed PDFs.

— Describes behavior but no flag or config to invoke it. v2.144.0

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261029 NOTES ↗

RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's latest snapshot adds organization- and workspace-level team permissions, easier GitHub repository and environment-variable management for deployments, clearer cancellation reporting, and new AI-powered trace search plus an in-page find — alongside breaking changes to the deployment Revisions tab and custom headers.

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

01 GitHub repository and environment variable management for deployments

IMPROVED 88

Deployments can now have their GitHub repository changed without creating a new deployment, via `Settings > Change repository` or by setting `source_config.repo_url` on `PATCH /v2/deployments/{id}` — useful when a repo is renamed or transferred. Environment variables on preview deployments can now also be edited and saved as a new revision using the same editor as regular deployments.

Point an existing deployment at a renamed or transferred GitHub repository without creating a new deployment.

```
$ curl -X PATCH
'https://api.smith.langchain.com/v2/deployments/<deployment-id>'
\
  -H 'X-API-Key: <your-api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"source_config": {"repo_url": "https://github.com/my-
org/new-repo-name"}}'
```

— Exact endpoint, field and UI path with runnable example snapshot-20261029

02 Cancellation reason reporting and reduced false thread error holds

IMPROVED 72

Cancelled runs now report why via a `cancel_reason` field, and thread error holds now only trigger on real failures rather than automation noise; threads can be filtered by

metadata (e.g. excluding `source: automation`) via `POST /threads/search` with `exclude_metadata`.

Filter out automation threads (e.g. Open SWE sidebar threads) when searching for user-initiated threads across a deployment.

```
$ curl -X POST 'https://api.smith.langchain.com/threads/search' \
  -H 'X-API-Key: <your-api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"exclude_metadata": [{"source": "automation"}]}'
```

— Named field and endpoint with a runnable curl example `snapshot-20261029`

THINNER COVERAGE BELOW

03 **In-page find for run content** NEW 30

Adds a Ctrl/Cmd+F Find feature for searching loaded run content.

— Bare description with no further mechanism or scope `snapshot-20261029`

└──▷ **BREAKING ON UPGRADE**

! The separate Revisions tab in Deployment overview is removed; bookmarks to tabs after Overview now open the tab to their right.

! Custom headers for a deployment are no longer shown in the Settings dialog; they now live only in the Connect dialog and are stored only in your browser.

! The Teams APIs return 403 unless the organization has `can_use_teams` enabled.

WAS THIS USEFUL? 👍 👎

Superlog Labs Superlog Feedback ⓘ

changes since 2026-09-29 CODE ↗

Superlog monitors AI agent telemetry with OpenTelemetry and groups signals into incidents for observability.

Superlog adds per-project Linear team selection and two ClickHouse ingest pacing options

└──▷ **TRY IT**

Fetch the Linear teams available to a project so you can populate a team-picker or validate a team id before patching automation settings.

```
$ curl -s
https://app.superlog.io/api/projects/<projectId>/linear/teams
```

Pin a project's agent tickets to a specific Linear team instead of always using the workspace's first team.

```
$ curl -s -X PATCH
https://app.superlog.io/api/projects/<projectId>/automation \
  -H 'Content-Type: application/json' \
  -d '{"linearDefaultTeamId": "<teamId>"}'
```

Slow down ClickHouse insert rate so rapid small batches merge into fewer, larger parts — useful when insert_quorum is enabled and part counts are climbing.

```
$ INGEST_CLICKHOUSE_MIN_INSERT_INTERVAL_MS=500
INGEST_CLICKHOUSE_KEEP_ALIVE=false node dist/worker.js
```

A new `GET /api/projects/:projectId/linear/teams` endpoint lists teams in a project's connected Linear workspace, returning `{ installed: false, teams: [] }` when no connection exists and 502 when Linear is unreachable.

A new `PATCH /api/projects/:projectId/automation` field `linearDefaultTeamId` sets the Linear team for agent tickets; `null` or a blank string clears it, and an id not found in the workspace is rejected with 400.

Control ClickHouse insert pacing with the `INGEST_CLICKHOUSE_MIN_INSERT_INTERVAL_MS` environment variable to set a

minimum time between inserts into the same table per worker, so faster inserts produce larger batches instead of more parts.

Control ClickHouse connection reuse with `INGEST_CLICKHOUSE_KEEP_ALIVE=false` to open a fresh connection per insert, spreading inserts evenly across replicas behind a TCP load balancer instead of drifting onto the slowest one.

The Settings page now shows a Default team picker in the File Linear ticket step, scoped to the project being edited, with Workspace's first team as the fallback option.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261029 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust shipped PagerDuty alerting for Loop automations, richer severity/category classification for Patterns findings, and a one-line way to trace Cursor agent sessions.

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

01 **Severity and category rubric for Patterns findings** IMPROVED 72

Patterns now assigns severity and categories to findings — such as `tool failure` or `safety and privacy risk` — and preserves manual corrections in later investigations. Teams can also customize the severity and category rubric Loop uses across a project's Patterns automations and conversations to match team-defined criteria (assignment requires data plane v2.18.0+, customization requires v2.19.0+).

— Explains mechanism and named requirements but no exact config surface.

snapshot-20261029

THINNER COVERAGE BELOW

02 **PagerDuty alerts for Loop automations** NEW 59

Loop automation alerts can now be sent to PagerDuty by setting an integration key and incident severity per destination; this requires data plane v2.17.0+.

— Names config needed but no exact UI path or API. snapshot-20261029

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI Red Teaming & Testing

AI-Infra-Guard

Feedback ⓘ v4.6.8 NOTES ↗ RANK

AI-Infra-Guard is an AI red teaming platform that scans AI infrastructure for CVE vulnerabilities, MCP/Agent skill risks, jailbreak weaknesses, and API relay security issues.

> AI-Infra-Guard v4.6.8 lets each scanner be installed and run independently via pip and adds a Runcrate provider preset for Agent Scan.

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

01 **Independently installable scanner packages** NEW 75

Each scanner can now be installed and run standalone via pip using the `aig-infra-scan`, `aig-model-scan`, `aig-mcp-scan`, `aig-skill-scan`, or `aig-agent-scan` packages, without starting the full web service.

Run the Agent scanner standalone against a Runcrate-backed agent without starting the full web service.

```
$ pip install aig-agent-scan
export LLM_API_KEY="your-api-key"
aig-agent-scan --agent_provider target.yaml -o result.json
```

— Names all five pip packages and a runnable install/run command. v4.6.8

02 **Runcrate provider preset for Agent Scan** NEW 60

Agent Scan now supports a Runcrate provider preset configured in `target.yaml` for testing Runcrate-backed agents.

Run the Agent scanner standalone against a Runcrate-backed agent without starting the full web service.

```
$ pip install aig-agent-scan
export LLM_API_KEY="your-api-key"
aig-agent-scan --agent_provider target.yaml -o result.json
```

— Names config file and preset but little mechanism detail. v4.6.8

WAS THIS USEFUL? 👍 👎

◆ AI/LLM Security

ToolHive

Feedback ⓘ v0.59.0 NOTES ↗ RANK

ToolHive runs MCP servers in isolated containers and enforces identity and access policies for local and Kubernetes deployments.

> ToolHive's v0.59.0 release focuses on authentication and client integration, adding TLS for operator-wide Redis defaults, a Windows credential helper for Claude Desktop, and several OAuth/token-exchange improvements.

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

01 **TLS for operator-wide Redis defaults** IMPROVED 45

ToolHive now enables TLS for operator-wide Redis defaults, securing Redis connections in Kubernetes operator deployments.

— Names the area but no config key or flag given. v0.59.0

02 **OAuth and token exchange improvements** IMPROVED 45

ToolHive now re-checks upstream Dynamic Client Registration (DCR) registrations while servers are running, keeping OAuth client metadata current without restarts; adds in-memory backend storage for exchange grant data to support OAuth token exchange grant persistence; and adds configured delegate client metadata so downstream OAuth clients receive accurate metadata from delegated authentication flows.

— Describes three auth-flow changes but no config keys or endpoints named. v0.59.0

03 **Windows credential helper for Claude Desktop** NEW 40

ToolHive adds a credential helper for Claude Desktop on Windows, extending MCP client integration to the Windows platform.

— States the capability but no setup steps given. v0.59.0

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

FrontiersMind Lumma-fev-27B-v2 Feedback ⓘ

released · Oct 29, 2026 MODEL ↗

Lumma-fev-27B-v2 from FrontiersMind was released on 28 Oct 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

NVIDIA-Nemotron-3.5-Transcribe-30B-A4B-DFlash2

Feedback ⓘ released · Oct 29, 2026 MODEL ↗

NVIDIA-Nemotron-3.5-Transcribe-30B-A4B-DFlash2 from NVIDIA was released on 7 Oct 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

inclusionAI Ling 3.1 Flash Feedback ⓘ

context change · Oct 28, 2026 MODEL ↗

Ling 3.1 Flash is a chat model from inclusionAI.

Context window grew from 262K to 1M tokens on OpenRouter (inclusionai/ling-3.1-flash), first seen 28 Oct 2026.

WAS THIS USEFUL? 👍 👎

Meta Llama 3.1 70B Instruct

Feedback ⓘ

retirement scheduled · Oct 28, 2026 MODEL ↗

Llama 3.1 70B Instruct is a chat model from Meta.

OpenRouter lists a retirement date of 3 Nov 2026 for meta-llama/llama-3.1-70b-instruct, first seen 28 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.6 27B

Feedback ⓘ

price increase · Oct 28, 2026 MODEL ↗

Qwen3.6 27B is a multimodal model from Alibaba.

Price increase on OpenRouter (qwen/qwen3.6-27b): $0.32 → $0.45 in, $3.20 → $2.70 out per million tokens (+41%), first seen 28 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.6 35B A3B

Feedback ⓘ

price cut · Oct 28, 2026 MODEL ↗

Qwen3.6 35B A3B is a multimodal model from Alibaba.

Price cut on OpenRouter (qwen/qwen3.6-35b-a3b): $0.10 → $0.10 in, $0.95 → $0.90 out per million tokens (-5%), first seen 28 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.8 27B

Feedback ⓘ price cut · Oct 28, 2026

MODEL ↗

Qwen3.8 27B is a multimodal model from Alibaba.

Price cut on OpenRouter (qwen/qwen3.8-27b): $0.20 → $0.039 in, $2.50 → $1.25 out per million tokens (-80%), first seen 28 Oct 2026.

WAS THIS USEFUL?  

◆ Frontier Models

Particle AI Particle-1

Feedback ⓘ

released · Oct 29, 2026 MODEL ↗

Particle-1 from Particle AI was released on 28 Oct 2026. 1049K-token context; $0.25 in / $1.00 out per million tokens (OpenRouter).

WAS THIS USEFUL?  

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 29, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue