THE AI TOOLCHAIN

NO. 008

Head

THE WEEK'S TOP PICKS

PUBLISHED OCTOBER 9, 2026 · FRIDAYS

// SHARE THIS WEEK ▽ Top 5 ▽ Top Pick

The best tooling updates that shipped this week.

▸ // HOW THESE PICKS ARE MADE SHOW

▾ // BENCHMARKS MOVED FOR TOOLS WE TRACK HIDE

Claude Code Terminal-Bench 4.0 Qwen3.8-Max-0902 `max`

27.0% −37.9 pts listed 2026-09-02 · leaderboard ↗

Claude Code Terminal-Bench 4.0 GLM-5.3-Flash `none`

35.8% −29.1 pts listed 2026-08-26 · leaderboard ↗

OpenAI Codex CLI Terminal-Bench 4.0 GPT-6 Luna `max`

16.4% −41.8 pts listed 2026-09-22 · leaderboard ↗

Claude Code Terminal-Bench 4.0 Sonnet 5.5 `max`

61.8% +3.9 pts listed 2026-09-28 · leaderboard ↗

OpenAI Codex CLI Terminal-Bench 4.0 GPT-6.1 Sol `max`

58.2% +0.0 pts listed 2026-09-29 · leaderboard ↗

Claude Code Terminal-Bench 4.0 Opus 5.5 `max`

64.8% +7.0 pts listed 2026-09-22 · leaderboard ↗

OpenAI Codex CLI Terminal-Bench 4.0 GPT-6 Sol `max`

49.4% −8.8 pts listed 2026-09-22 · leaderboard ↗

---

eek Models APEX-Agents DeepSeek-V3.2 `Thinking; Novita`

vim ? help

21.3% −26.0 pts released 2025-12-01 · leaderboard ↗

GLM APEX-Agents GLM-5.2

45.2% −11.4 pts released 2026-06-16 · leaderboard ↗

Nemotron APEX-Agents nemotron-3-ultra

22.7% +11.2 pts released 2026-06-04 · leaderboard ↗

Grok API APEX-Agents Grok 4.7

54.6% +39.4 pts released 2026-09-21 · leaderboard ↗

Gemini API APEX-Agents Gemini 3.5 Flash-Lite

29.3% −38.5 pts released 2026-07-21 · leaderboard ↗

Scores from Epoch AI Benchmarking Hub (CC BY 4.0), evaluated 2026-10-02 to 2026-10-09.

▾ // LOCAL LLM HARDWARE THIS WEEK HIDE

ASUS Ascent GX10 list price changed **$5,999** -14% 2026-10-08 · source ↗

Vendor list prices, captured live and via the Wayback Machine; MSRP for cards until street capture ships. Events dated 2026-10-02 to 2026-10-09; each device page carries the full history.

▾ // THE BRIEF HIDE

What stands out in this week's picks, grouped by what it lets you do. Every tool named links to its pick below.

Three unrelated things stand out. vLLM's `vllm preload` keeps weights warm in a daemon so a server restart no longer means re-reading hundreds of gigabytes from disk. Docker Sandboxes and Pipelock both put agent execution inside a network-isolated box you can actually inspect. Hayhooks checkpoints pipeline runs to Redis, so a crash mid-run resumes instead of restarting.

> DEPLOY
>
> **Restart an inference server without paying the cold-load tax again**
>
> vLLM v0.31.0's `vllm preload` daemon holds model weights in a cache across process restarts, turning a config change or crash recovery from a multi-minute weight load into a fast relaunch. SGLang v0.5.21 adds live prefill-decode role switching with KV transfer checksums, so a

disaggregated deployment can rebalance which nodes do prefill versus decode without draining and redeploying, and detect a corrupted KV transfer rather than serving garbage from it.

vLLM · SGLang

GOVERN

Give an agent a shell without giving it your network

Docker Cloud Sandboxes run agent workloads in isolated microVMs driven by an `sbx` CLI, with an open spec for packaging the environment so the same sandbox definition is reproducible across a team. Pipelock v3.6.0 takes the adjacent route — network namespace isolation plus loopback service declarations means an agent reaches only the services you enumerated, and signed session contracts plus VNC operator control give you a record and a live view of what it did.

Docker Sandboxes · Pipelock

OPERATE

Survive a restart in the middle of a long pipeline run

Hayhooks 2.0 checkpoints pipeline state to Redis and streams over reattachable SSE, so a deploy or a dropped connection mid-run resumes from the last checkpoint rather than re-running every upstream step. The same machinery makes human-in-the-loop suspension practical: a run can park waiting on an approval for as long as it needs instead of holding a request open.

hayhooks

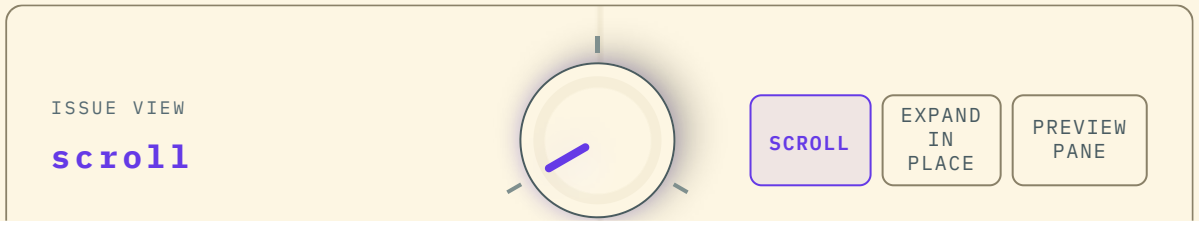
BUILD

Get a calibrated probability out of a model instead of parsing prose

Cloudflare's Clef models on Workers AI answer questions as structured probabilities rather than free text, with open weights, image input and a fine-tuning service. For routing, triage and classification work, that removes the usual layer of regexes and retry logic that exists only to coerce a chat model into a decision you can threshold on.

Cloudflare Workers AI

VIEW



INFORMATION DENSITY

everything



BRIEFED | **EVERYTHING**

Every change in this week's picks, in full.

all 53 changes

Do you prefer **scroll**? 👍 👎

// THE PICKS

01	**Cloudflare Workers AI**	Agentic Infrastructure
02	**Composio**	MCP Servers & Tooling
03	**Docker Sandboxes**	AI/LLM Security
04	**vLLM**	Local LLM Runtimes
05	**Pipelock**	AI/LLM Security

01 Cloudflare Workers AI

Workers AI - Introducing Clef: Cloudflare's first open-source decision models, now on Wo

Agentic Infrastructure

Cloudflare Workers AI runs open-weight models, including the Clef decision models, as serverless inference on Cloudflare's global network.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 5/5 NOVELTY 5/5

Cloudflare's first trained models ship as open-weight (Apache 2.0) decision models that return structured probabilities — yes/no, single-choice, and rubric scores — instead of free-form text, with visual classification over up to four images and System One API compatibility as a drop-in Jev replacement.

> Cloudflare launched Clef, its first open-source decision models on Workers AI, adding structured probability-based question answering, image inputs, a Jev-compatible migration path, open weights, and a fine-tuning service.

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

01 **Structured question types for decisions** NEW 95

Clef introduces three structured question types for a single request: `noul` (yes/no probability), `choice` (pick one from a defined set with per-option probabilities), and `score` (probability-weighted rubric rating), with up to 64 questions supported per request, returning structured probabilities instead of free-form text.

Route a support ticket in the hot path — determine urgency and owning team in one structured call, with no free-form output to parse.

```
$ curl
https://api.cloudflare.com/client/v4/accounts/{account_id}/ai/run
/@cf/cloudflare/clef-flash \
  -H 'Authorization: Bearer <API_TOKEN>' \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "clef-flash",
    "state": "Checkout has been failing for every customer for
the last hour.",
    "questions": {
      "urgent": {
        "type": "noul",
        "instructions": "Is this support request urgent?"
      },
      "team": {
        "type": "choice",
        "instructions": "Which team should handle this request?",
        "criteria": {
          "billing": "Payments, invoices, and refunds",
          "technical": "Outages, errors, and configuration",
          "sales": "Plans and upgrades"
        }
      }
    }
  }'
```

Score a user submission against a trust-and-safety policy rubric before taking automated action — using the `score` question type inside a Worker.

```javascript
const response = await env.AI.run('@cf/cloudflare/clef', {
  model: 'clef',
  state: userSubmissionText,
  questions: {
    policy_compliance: {
      type: 'score',
      instructions: 'How well does this submission comply with
our content policy?',
      criteria: {
        1: 'Clear violation',
        2: 'Likely violation',
        3: 'Borderline',
        4: 'Compliant',
        5: 'Fully compliant'
      }
    }
  }
});
// response.answers.policy_compliance.score -> probability-
weighted score
// Act on score threshold without parsing free-form text
```

— Names all three question types, the 64-question limit, and runnable examples

Workers AI - Introducing Clef: Cloudfla…

02 **Clef and Clef Flash decision models launch** NEW 75

Cloudflare launched `@cf/cloudflare/clef` (27B parameters, 64K-token context) and `@cf/cloudflare/clef-flash` (9B parameters, 64K-token context), described as Cloudflare's first trained models. Both are available via env.AI.run() in Workers and the REST API at `/ai/run`.

— Names both model ids, sizes, context length and access methods

Workers AI - Introducing Clef: Cloudfla…

03 **System One API for Jev-compatible migration** NEW 65

Clef implements the System One API, making it a drop-in replacement for Jev integrations by changing only the endpoint and model identifier.

— Gives a concrete migration path though no exact endpoint shown

Workers AI - Introducing Clef: Cloudfla…

THINNER COVERAGE BELOW

04 **Open model weights on Hugging Face** NEW 35

Cloudflare released model weights for both `clef` and `clef-flash` under the Apache 2.0 license on Hugging Face.

— Names license and host but no further mechanism

Workers AI - Introducing Clef: Cloudfla…

05 **Reinforcement learning fine-tuning service** NEW 35

A new reinforcement learning fine-tuning service lets teams tune Clef for custom workloads; it currently requires design-partner sign-up.

— Names the service but access is gated and mechanism unclear

Workers AI - Introducing Clef: Cloudfla…

06 **AI Gateway compatibility for Clef** NEW 35

Clef is compatible with AI Gateway via the same env.AI.run() and `/ai/run` endpoints used for direct calls.

— Thin note reusing endpoints already named elsewhere

Workers AI - Introducing Clef: Cloudfla…

07 **Visual classification with image inputs** NEW 30

Clef accepts up to four images alongside the text state for classification, unlike text-only decision models.

— States the capability but no example or mechanism detail

Workers AI - Introducing Clef: Cloudfla…

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

02 Composio

announced · Oct 8, 2026

MCP Servers & Tooling

Composio powers 1000+ toolkits, tool search, context management, authentication, and a sandboxed workbench to help you build AI agents that turn intent into action.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 5/5 NOVELTY 5/5

Agents can now reach 100+ premium tools (Apollo, Exa, Parallel, People Data Labs) on a pay-per-call basis through a built-in wallet, with no per-provider signups or API keys to manage. Per-call charges surface in each tool response so developers can track or pass through the cost.

> Composio launched Composio Instant, letting agents pay per call for 100+ premium third-party tools from a built-in wallet without provider signups or API keys.

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

01 Composio Instant pay-per-call wallet for 100+ tools NEW

93

Composio Instant gives agents wallet-backed, pay-per-call access to over 100 premium third-party tools (Apollo, People Data Labs, Exa, Parallel, Higgsfield, ZeroBounce, and others) alongside user-connected apps, with no provider account creation or subscription required. Enable it via the `instant` parameter on composio.sessions.create() (Python) or composio.create() (TypeScript), using `instant.toolkits.enable` to pick which toolkits run on the wallet; set `instant.returnInstantCharge` (`return_instant_charge` in Python) to get an `instant_charge` field (`instantCharge` in TypeScript) back on each Instant tool call response for cost tracking or pass-through billing. Every account receives $2 of Instant credit at signup, Pro accounts get $29/month, and failed calls are never charged; it can also be toggled on per-project from the dashboard under Composio Instant, with available apps browsable under Composio For You.

Give an agent pay-per-call access to Parallel (Instant) while using the user's own Gmail connection, and capture the per-call charge in each response for billing.

```python
from composio import Composio

composio = Composio()
session = composio.sessions.create(
    user_id="user_123",
    toolkits=["parallel", "gmail"],
    instant={
        "toolkits": {
            "enable": ["parallel"]
        },
        "return_instant_charge": True,
    },
)
tools = session.tools()
# Each Instant tool response now includes 'instant_charge'
```

Same flow in TypeScript: enable Instant for Parallel with charge reporting so your app can surface or pass through per-call costs to end users.

```typescript
import { Composio } from "@composio/core";

const composio = new Composio();
const session = await composio.create("user_123", {
  toolkits: ["parallel", "gmail"],
  instant: {
    toolkits: { enable: ["parallel"] },
    returnInstantCharge: true,
  },
});
const tools = await session.tools();
// Each Instant tool response includes 'instantCharge'
```

Enable Composio Instant for a project from the dashboard so agents can immediately start spending wallet credit on supported tools.

📍 1. Open your Composio dashboard. 2. Go to your project settings. 3. Find `Composio Instant` and toggle it on. 4. Browse available Instant apps at `Composio For You` to confirm which providers are wallet-enabled.

— Full mechanism, exact SDK fields, credit amounts, and runnable examples given.

`launch-20261008-ac841e8a`

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

03 Docker Sandboxes

announced · Oct 1, 2026

AI/LLM Security

Docker Sandboxes isolates containerized applications in secure, lightweight environments for testing and development without affecting the host system.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 4/5 NOVELTY 4/5

Docker Cloud Sandboxes run agent tasks in isolated microVMs on Docker-managed compute and can migrate a sandbox's filesystem from a laptop to the cloud mid-task so a long run continues after you close the lid; the accompanying Sandbox Kit OCI spec lets teams build, scan, and pin agent environments by digest.

> Docker launched Docker Cloud Sandboxes, giving teams isolated microVMs for agent workloads via a new `sbx` CLI, alongside a new open specification for packaging agent environments and supporting configuration tooling.

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

01 Docker Sandbox Kit specification NEW

65

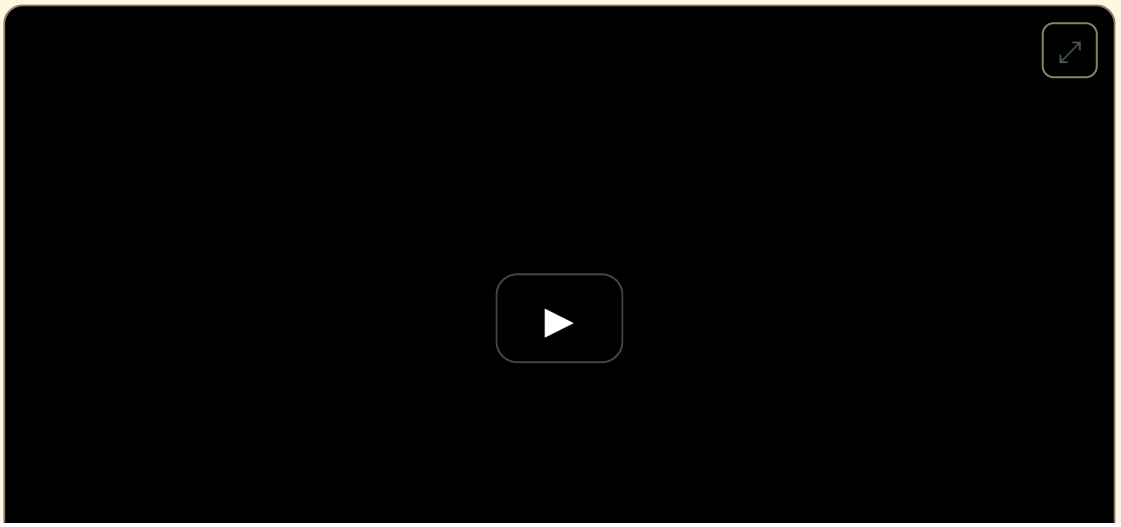
Introduces the Docker Sandbox Kit specification, an OCI image format containing the agent, its tools, and access declarations, enabling teams to build, push, pull, scan, and pin agent environments by digest. The spec is published under Apache 2.0 with a commitment to bring it to the CNCF for neutral governance, and Docker Sandboxes is the first runtime to implement it.

— Explains format and governance but no runnable command shown. launch-20261001-8151baca

02 Docker Cloud Sandboxes with sbx CLI NEW

60

Run agent tasks in isolated microVMs on Docker-managed cloud compute using the `sbx` CLI. Supports moving a sandbox's filesystem from a local machine to the cloud mid-task, allowing longer agent runs to continue after closing a laptop.









— Names sbx CLI and microVM mechanism but no exact command syntax.

launch-20261001-8151baca

THINNER COVERAGE BELOW

03 **sbxenv configuration files** NEW 50

Introduces `sbxenv` files to describe a set of sandbox environments as configuration, enabling reproducible AI-ready developer environments across machines.

— Names config file surface but lacks schema detail. launch-20261001-8151baca

04 **MCP tool support in sandboxes** NEW 40

Supports tools connected through Model Context Protocol (MCP) inside sandboxes, including network rules and Kit packaging for agent workflows.

— Named protocol and rules but mechanism left thin. launch-20261001-8151baca

---

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

04 vLLM v0.31.0

Local LLM Runtimes

A high-throughput and memory-efficient inference and serving engine for LLMs

// WHY IT MADE THE LIST — DEPTH 5/5 · IMPACT 4/5 · NOVELTY 3/5

A new vllm preload daemon keeps post-quantized weights resident in GPU memory across engine restarts, cutting cold-start cost for RL loops and frequent redeploys, alongside expanded speculative decoding and a per-request multimodal security gate.

vLLM v0.31.0 ships a `vllm preload` fast-restart weight-cache daemon, new admission and MoE-serving flags (`--max-num-active-seqs` , `--all2all-backend moonep`), broad speculative-decoding advances on Model Runner V2, and several breaking changes including mandatory opt-in for per-request multimodal kwargs, a renamed mamba prefix-cache flag, and removal of the slow tokenizer mode and legacy fp8 quantization path.

└──▷ **WHAT SHIPPED · 20 FEATURES** most completely described first ? what's the number?

01 Cap on concurrently active sequences NEW

83

Adds `--max-num-active-seqs` flag to cap RUNNING-queue admission independently of `max_num_seqs` , letting operators limit concurrent in-flight sequences separately from the total sequence limit.

Cap the number of concurrently active sequences to prevent memory pressure from new admissions on a shared inference node.

```
$ vllm serve meta-llama/Llama-3-8B --max-num-active-seqs 64 --max-num-seqs 256
```

— Flag and example command fully specified v0.31.0

02 vllm preload CLI for fast-restart weight caching NEW

80

Adds a new `vllm preload` CLI subcommand that launches a weight-cache daemon keeping post-quantized weights resident in GPU memory across engine restarts, with data-parallelism support, MTP draft model support, a `/health` endpoint, and a readiness wait.

— Names command and endpoint but no example invocation shown v0.31.0

03 **MoonEP all-to-all backend for MoE serving** NEW 78

Adds `--all2all-backend moonep` flag to enable the MoonEP balanced expert-parallel all-to-all backend for large-scale MoE serving.

Route MoE all-to-all traffic through the MoonEP balanced backend for large expert-parallel deployments.

```
$ vllm serve deepseek-ai/DeepSeek-V4.1-Flash --all2all-backend
moonep --tensor-parallel-size 8
```

— Flag named with runnable example command v0.31.0

04 **Trust gating for per-request multimodal kwargs** BREAKING 78

Adds `--trust-request-mm-kwargs` flag; per-request `mm_processor_kwargs` and `media_io_kwargs` are now rejected unless this flag is set, hardening the multimodal attack surface. Existing deployments that pass per-request multimodal kwargs will break without this flag enabled.

— Flag named with clear migration behavior, no example v0.31.0

05 **Speculative decoding advances on Model Runner V2** NEW 68

Adds the LiLiCorr speculative-decoding drafter, draft-model speculative decoding with custom logits processors validated at admission, DFlash async scheduling with context K/V precompute captured in the draft CUDA graph, DSpark adaptive verification for Gemma4, variable-length decode for Kimi-K3, and Cohere2MoE auxiliary hidden states for EAGLE3/DFlash drafters, all on Model Runner V2.

— Names six distinct drafters/models but no usage surface given v0.31.0

06 **Quantization backend changes: fp8_per_tensor and removals** 63

BREAKING

Replaces the online quantization path `quantization="fp8"` with the `fp8_per_tensor` shorthand; the old form no longer works. Also removes Quark silent online quantization and the AllSpark INT8 W8A16 backend.

— Names old and new quantization values and two removed backends v0.31.0

07 **LoRA support for new model types** NEW 60

Adds LoRA support for Nemotron VL language models, ModernBert, VoyageQwen3 embeddings, and RoBERTa sequence classification, plus `modules_to_save` sequence-classification heads with per-adapter `num_labels`.

– Names five model/LoRA targets and config field, no example v0.31.0

THINNER COVERAGE BELOW

08 **Large-scale serving parallelism and offloading improvements** NEW 58

Adds prefill context parallelism combined with data parallelism for large-scale serving, DeepEPv2 with sequence parallelism and EPLB with shared-expert overlap, a sharding-aware NCCL M2N weight-transfer backend for reinforcement-learning workloads, and KV offloading back-pressure detection.

– Names four distributed-serving mechanisms, no config surface given v0.31.0

09 **Mamba prefix-cache flag renamed** BREAKING 58

`--enable-mamba-fine-grained-prefix-cache` is renamed to `--enable-mamba-shared-prefix-checkpoint`; scripts using the old flag name will break.

– Names both old and new flag exactly v0.31.0

10 **Adaptive long-prefill chunking threshold** IMPROVED 55

The `--long-prefill-token-threshold` scheduler behavior now adapts the threshold to the number of waiting prefills instead of always chunking a lone request.

– Names flag, describes behavior change, no example v0.31.0

11 **Prefix-cache key isolation by source** IMPROVED 55

Adds prefix-cache extra-key tagging by source so a LoRA name and a `cache_salt` can no longer collide; the LoRA path is now part of the block hash.

– Names cache_salt field and mechanism, no flag given v0.31.0

12 **XPU CUDA graphs enabled by default** BREAKING 55

Enables XPU CUDA graphs by default, previously opt-in via the `VLLM_XPU_ENABLE_XPU_GRAPH` environment variable, which has now been removed.

– Names removed env var and behavior change v0.31.0

13 **Removal of slow tokenizer mode** BREAKING 52

`tokenizer_mode="slow"` has been removed; any configuration using this value will fail.

– Names exact config value removed and failure behavior v0.31.0

14 **Enforce-eager disables JIT kernel warmup** BREAKING 50

`--enforce-eager` now also disables JIT kernel warmup, which may affect startup behavior for users who relied on JIT warmup under eager mode.

– Names flag and behavior change but no migration step v0.31.0

15 **Fill-in-the-middle completion for DeepSeek V4** NEW 49

Adds FIM (fill-in-the-middle) completion via the `suffix` field for DeepSeek V4.

– Names field but no request example v0.31.0

16 **KV-block-aware scheduling priority** IMPROVED 47

Reworks the waiting queue so requests already holding KV blocks are scheduled first, reducing recomputation under memory pressure.

– Explains mechanism but no named flag or config v0.31.0

17 **Granite 4.2 reasoning parser** NEW 40

Adds the `granite_thinking_parser` built-in reasoning parser for Granite 4.2.

– Names parser but no usage detail v0.31.0

18 **MiMo V2 model support additions** NEW 40

Adds MiMo V2 MXFP4 MoE, a BF16 MoE router, and DFlash draft support.

– Names three MiMo V2 additions, no usage detail v0.31.0

19 **Encoder CUDA graphs for vision towers** NEW 38

Adds encoder CUDA graphs for the vision tower on DeepSeek-V4.1-Flash and other multimodal models.

– Names target model, no mechanism detail beyond CUDA graphs v0.31.0

20 **Structured generation mode for DiffusionGemma** NEW 35

Adds DiffusionGemma structured generation mode with bounded single-token choices.

– Thin description, no config surface given v0.31.0

└──▷ **BREAKING ON UPGRADE**

! Per-request `mm_processor_kwargs` and `media_io_kwargs` are now rejected unless `--trust-request-mm-kwargs` is set; existing deployments that pass per-request multimodal kwargs will break without this flag.

! `tokenizer_mode="slow"` has been removed; any configuration using this value will fail.

! `--enable-mamba-fine-grained-prefix-cache` is renamed to `--enable-mamba-shared-prefix-checkpoint`; scripts using the old flag name will break.

- ! Online quantization via `quantization="fp8"` is replaced by the `fp8_per_tensor` shorthand; the old form no longer works.
- ! Quark silent online quantization has been removed.
- ! The AllSpark INT8 W8A16 backend has been removed.
- ! `--enforce-eager` now also disables JIT kernel warmup, which may affect startup behavior for users who relied on JIT warmup under eager mode.
- ! `VLLM_XPU_ENABLE_XPU_GRAPH` environment variable has been removed; XPU graphs are now always enabled by default on XPU.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

05 Pipelock

announced · Oct 6, 2026

AI/LLM Security

Pipelock is an open-source AI agent firewall that scans MCP and HTTP egress for exfiltration, SSRF, and prompt injection and emits signed action receipts.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 4/5 NOVELTY 4/5

Contained agents now each run in their own network namespace with a verified session contract, and pipelock contain view opens a contained agent's screen over VNC with single-controller takeover; signed workspace change statements record exactly what each session altered.

> Pipelock v3.6.0 ships a full contained-agent environment — network namespace isolation, VNC operator control, loopback service declarations, and signed session contracts — alongside a large hardening pass across redirect handling, receipt durability, DLP, and config validation that introduces numerous breaking changes.

└──▷ WHAT SHIPPED · 21 FEATURES most completely described first ? what's the number?

01 Contained-agent network namespace, VNC control and session contracts

NEW 98

v3.6.0 expands the containment module with `containment.published_services` to publish contained-agent services to the operator, `containment.display` / `containment.display.geometry` to provision a screen for a contained browser, `containment.loopback_services` to declare extra local destinations (host, port, owner, reason, expiry) with privileged timer removal on expiry and `contain verify` enforcement, `pipelock contain view [--control]` to watch or take over the contained screen via a caller-only local VNC socket (view-only sessions cannot type or click; only one controller at a time), `pipelock contain run --dry-run` to print the session contract (agent user, egress posture, capsule path, private `/tmp` state, tools and workspace grants with expiry) without launching, `pipelock posture verify --workspace-statement` to check the signed workspace change statement against the posture capsule (exits 2 on an incomplete statement), private `/tmp` / `/var/tmp` via a transient systemd service, and `contain install` adding the interception CA to the contained agent's Chromium certificate database (reported as probe 20).

Preview the full session contract for a contained agent before launch, catching expired grants in CI without starting the agent.

```
$ pipelock contain run --dry-run --config ./pipelock.yaml
```

Declare a local service (e.g. a code-indexing daemon) that a contained agent is allowed to reach, with an expiry so the exemption is automatically removed.

```yaml
containment:
  loopback_services:
    - host: 127.0.0.1
      port: 7701
      owner: code-indexer
      reason: 'Agent requires local code-indexing service'
      expiry: '2026-12-31T00:00:00Z'
```

Watch a contained agent's screen live to audit its interactive behaviour, or pass `--control` to take over the session.

```
$ pipelock contain view --control
```

— Names every config key and ships runnable commands for each surface. v3.6.0

---

02 **Redirect hop mediation in absolute-URI forward proxy** BREAKING

83

The absolute-URI forward proxy now returns allowed redirects (301, 302, 303, 307, 308) to the client instead of following them internally, so each hop is independently admitted, receipted and mediation-enveloped; a redirect whose Location carries a backslash, whitespace, control character or is duplicated is blocked with 403 and reason `ambiguous redirect location`. The `redirect` audit event message changes from `redirect followed` to `redirect observed`.

— Named status codes and block reason, but no runnable example. v3.6.0

---

03 **Enforced maximum expiry horizons for temporary exceptions**

80

BREAKING

Temporary exception fields `request_body_scanning.sigv4_credential_routes[].expires`, `response_scanning.core_observe_exceptions[].expires`, `sandbox.best_effort_expiry`, `request_body_scanning.content_entropy_warn_routes[].expires`, `response_scanning.unscannable_passthrough[].expires`,

`fetch_proxy.monitoring.path_entropy_exclusions[].expires`, and `fetch_proxy.monitoring.query_entropy_param_exclusions[].expires` now have enforced maximum expiry horizons of 30, 30, 30, 90, 90, 180, and 180 days respectively, checked at load and reload; a config-sourced sandbox expiry must be an absolute RFC3339 time.

— Enumerates every field and its exact day limit. v3.6.0

---

04 **Trusted hosts for request-body scanning exceptions** BREAKING

74

Adds `request_body_scanning.trusted_hosts` to let request-body injection and fully redacted critical credentials follow the configured action rather than a hard block for listed destinations; this list is now the authoritative source for request-body trust, so operators who relied on `response_scanning.exempt_domains` to relax outbound request bodies must add those hosts here instead.

— Names both config keys and the migration step. v3.6.0

---

05 **Deferred-resolution source tracking for kill switch and upstream gate**

73

BREAKING

Activating the kill switch now blocks held MCP approvals that have not yet started sending, recording source `kill_switch` in the journal and layer `kill_switch` in the signed receipt (a previously-forwarding held call after activation now resolves as blocked with error `-32004`). Separately, a held HTTP call refused by the upstream gate at release now records resolution source `upstream_contract` in deferred-resolution records.

— Names error code and sources but gives no direct reader action. v3.6.0

---

06 **Receipt durability enforcement for credential and MCP allows**

73

BREAKING

With `flight_recorder.require_receipts`, a credential-audience, issuer-cookie or issuer-query allow now forwards only after its receipt is durably written; a failed or unsupported write blocks the request with reason `receipt_emission_failed` and layer `credential_audience_receipt` or `issuer_allow_receipt`, and the same durability requirement now applies generally whenever MCP receipts are required.

— Names the config key and failure reasons but no setup steps. v3.6.0

---

07 **GitLab credential header scanning** NEW

73

GitLab Private-Token and Job-Token headers are now scanned for credentials by default, with `dlp.gitlab_hosts` available to declare self-managed GitLab hosts

whose tokens are treated as going to their issuer.

— Names both headers and the config key for self-managed hosts. v3.6.0

---

08 **Verification mode reporting for offline audit packets** 72

BREAKING

`audit-packet --offline` no longer reports a packet as trusted; the Go, Rust and TypeScript verifiers now return `verdict: schema_checked_trust_unverified` with `trusted` and `valid` false and exit non-zero, and `pipelock verify-receipt --clean-report` output now requires a `verification_mode` field (`pinned_provenance` or `unpinned_structural`).

— Names the exact verdict string and new required field. v3.6.0

---

09 **Per-process receipt chains** BREAKING 65

Each Pipelock process now writes its own receipt chain rather than a single shared session file; `verify-receipt --chain` and `evidence doctor` follow restart links between chains to reconstruct the full history.

— Names the two verification commands but not full linking mechanics. v3.6.0

---

10 **Scan API warn decision** BREAKING 62

The Scan API can now return `decision: "warn"` for a `tool_call` matching a warn-configured tool-policy rule, aligning with live MCP proxy behaviour; consumers that expected only `allow` or `deny` must now handle the new value.

— Names the field and value but no example payload. v3.6.0

---

11 **Named-agent policy enforcement (Enterprise)** BREAKING 61

Named agent policy (Enterprise) now follows bound identity only; a self-declared `X-Pipelock-Agent` header is recorded for attribution but receives the fallback policy, and session state folds to the client IP. `contain install` now also refuses to swap in a core build that cannot enforce a named-agent config.

— Names the header and fallback behaviour but is Enterprise-only. v3.6.0

---

12 **Shipped egress blocklist applied by default** BREAKING 60

A config file that omits `fetch_proxy.monitoring.blocklist` now gets the shipped blocklist instead of an empty one, so destinations on the shipped list that were reachable under 3.5.0 will now be blocked; set `blocklist: []` to keep it off.

— Names the config key and exact opt-out value. v3.6.0

THINNER COVERAGE BELOW

13 **SSH public key exemption from Credential File Access rule**

IMPROVED

56

Reading an SSH public key (`id_*.pub` under `.ssh`) no longer trips the built-in Credential File Access rule when the path is the call's only string value, across the MCP proxy, `pipelock explain` , the Scan API and hook events.

— Names the rule and exact surfaces affected but no example call. v3.6.0

14 **TLS passthrough wildcard restriction over public suffixes**

55

BREAKING

`tls_interception.passthrough_domains` now refuses a wildcard over a public suffix, including private-section suffixes shared by many tenants; operators must use exact hosts or intercept with a trusted local CA.

— Names the config key and the required workaround. v3.6.0

15 **Host pattern validation across allow/deny/trust/bypass lists**

54

BREAKING

Host patterns in allow, deny, trust and bypass lists are now validated and refused if they carry a URL, `host:port` , fragment, interior wildcard, malformed label, or a trailing extra dot.

— Lists rejected pattern forms but no example rewrite. v3.6.0

16 **GitHub release-asset JWT allowed in default policy** NEW

52

The default canonical policy hash changes because a signed GitHub release-download JWT may now reach `release-assets.githubusercontent.com` in an `https` URL query; receipt consumers that pin the default hash will see a new value.

— Names the host and mechanism but not the triggering rule. v3.6.0

17 **session_profiling.volume_spike_ratio removed** DEPRECATED

48

`session_profiling.volume_spike_ratio` is removed; a config that still sets it, at top level or per agent, is refused at load, and the canonical policy hash changes as a result.

— States removal and effect but no replacement guidance. v3.6.0

18 **systemd reload verdict wait** IMPROVED 43

`systemctl reload pipelock` now waits for the reload verdict on systemd 253 and newer, using `Type=notify-reload`.

— States the mechanism with no further behaviour detail. v3.6.0

19 **Preset tool-policy rule arg_source** BREAKING 43

Shipped presets now set `arg_source: patch_targets` on tool-policy rules; binaries older than 3.6.0 refuse to load them.

— Names the field and compatibility impact only. v3.6.0

20 **Custom DLP pattern exempt_domains restriction** BREAKING 43

A custom DLP pattern that reuses a core pattern name can no longer carry `exempt_domains`; config validation refuses it at startup and reload.

— Names the field and validation point only. v3.6.0

21 **file_sentry.action block refused on server listener** BREAKING 43

`pipelock run` now refuses `file_sentry.action: block` on the server listener; a reload requesting it is rejected atomically.

— Names the config value and refusal point only. v3.6.0

└──▷ BREAKING ON UPGRADE

! A config file that omits `fetch_proxy.monitoring.blocklist` now gets the shipped blocklist instead of an empty one; destinations on the shipped list that were reachable under 3.5.0 will now be blocked. Set `blocklist: []` to keep it off.

! The `request_body_scanning.trusted_hosts` list is now the authoritative source for request-body trust; operators who relied on `response_scanning.exempt_domains` to relax outbound request bodies must add those hosts to `request_body_scanning.trusted_hosts`.

! Host patterns in allow, deny, trust and bypass lists are now validated and refused if they carry a URL, `host:port`, fragment, interior wildcard, malformed label, or a trailing extra dot.

! `tls_interception.passthrough_domains` now refuses a wildcard over a public suffix (including private-section suffixes shared by many tenants); use exact hosts or intercept with a trusted local CA.

! Temporary exception expiry limits are now enforced at load and reload: `request_body_scanning.sigv4_credential_routes[].expires` and `response_scanning.core_observe_exceptions[].expires` (30 days), `sandbox.best_effort_expiry` (30 days), `request_body_scanning.content_entropy_warn_routes[].expires` and

`response_scanning.unscannable_passthrough[].expires` (90 days), `fetch_proxy.monitoring.path_entropy_exclusions[].expires` and `fetch_proxy.monitoring.query_entropy_param_exclusions[].expires` (180 days). A config-sourced sandbox expiry must be an absolute RFC3339 time.

! `session_profiling.volume_spike_ratio` is removed; a config that still sets it (top-level or per agent) is refused at load. The canonical policy hash changes as a result.

! A custom DLP pattern that reuses a core pattern name can no longer carry `exempt_domains`; config validation refuses it at startup and reload.

! `pipelock run` now refuses `file_sentry.action: block` on the server listener; a reload requesting it is rejected atomically.

! Named agent policy (Enterprise) now follows bound identity only; a self-declared `X-Pipelock-Agent` header is recorded for attribution but receives the fallback policy, and session state folds to the client IP.

! `contain install` now refuses to swap in a core build that cannot enforce a named-agent config.

! Each Pipelock process now writes its own receipt chain; tools that read a single session file must read the run chains instead. `verify-receipt --chain` and `evidence doctor` follow the links.

! Activating the kill switch now blocks held MCP approvals that have not yet started sending; a held call that previously forwarded after activation now resolves as blocked with error `-32004`.

! The default canonical policy hash changes because a signed GitHub release-download JWT may now reach `release-assets.githubusercontent.com` in an `https` URL query; receipt consumers that pin the default hash will see a new value.

! The Scan API can now return `decision: "warn"`; consumers that expected only `allow` or `deny` must handle the new value.

! Shipped presets now set `arg_source: patch_targets` on tool-policy rules; binaries older than 3.6.0 refuse to load them.

! Contained hosts must run `pipelock contain install` with the new binary to adopt the private network namespace, private temporary directories, `0600` config mode and refreshed CA export.

! The absolute-URI forward proxy now hands allowed redirects back to the client instead of following them; clients that do not follow redirects themselves will no longer have hops followed on their behalf.

! The `redirect` audit event message changes from `redirect followed` to `redirect observed`; anything matching on the old message text must be updated.

! With `flight_recorder.require_receipts` enabled, a credential-audience, issuer-cookie or issuer-query allow is now blocked if its receipt write fails, with reason `receipt_emission_failed`.

! When MCP receipts are required, a request now forwards only after its receipt is durably written; a failed or unsupported receipt write blocks the request.

! `audit-packet --offline` no longer reports a packet as trusted; the Go, Rust and TypeScript verifiers return `verdict: schema_checked_trust_unverified` with `trusted` and `valid`

false and exit non-zero. `pipelock verify-receipt --clean-report` output now requires a `verification_mode` field (`pinned_provenance` or `unpinned_structural`).

! GitLab `Private-Token` and `Job-Token` headers are now scanned for credentials by default; self-managed GitLab operators must add their host to `dlp.gitlab_hosts` so tokens are treated as going to their issuer.

! Reading an SSH public key (`id_*.pub` under `.ssh`) no longer trips the built-in Credential File Access rule when the path is the call's only string value on the MCP proxy, `pipelock explain`, the Scan API and hook events.

announced · Oct 6, 2026

Pipelock 3.6 adds per-agent network namespaces, VNC screen viewing/control, signed workspace change statements, per-process receipt chains, and `verify-receipt --whole-recorder`.

COVERAGE

UPDATED detections

Partial-match detection for canary, environment, and file secrets, SSRF and allowlist/blocklist checks extended to URL destinations in query parameters, Gzip and deflate response decoding before scanning, Issuer-bound credential routing for built-in credential classes

NEW detection DNS-over-HTTPS message inspection

NEW integrations

Pi listener integration via `pipelock pi install`, Continue.dev MCP server wrapping via `pipelock continue install`, Hand-setup guide for Grok CLI

All of Pipelock's coverage changes →

› Contained agents run in their own network namespace with their own loopback and no external route; `contain run` verifies the namespace before every launch.

› `pipelock contain view` opens a contained agent's screen via VNC through a local-only socket; `--control` enables single-controller keyboard/mouse takeover.

› `contain run` prints a session contract before launch (agent user, egress posture, private `/tmp`, allowed tools, workspace grants with expiry); `--dry-run` prints and stops.

› When a contained session with granted workspaces ends, `contain run` writes a signed workspace change statement listing added, removed, and modified paths bound to the session's posture capsule.

› Each Pipelock process now writes its own receipt chain; `verify-receipt --whole-recorder` checks all chains and shutdown seals on disk in one pass.

+4 more

└─▷ **BREAKING ON UPGRADE**

- ! A host pattern that is not a valid hostname is now refused with a message naming the field.
- ! A wildcard passthrough over a public suffix is now refused.
- ! A temporary exception whose expiry exceeds the field's new maximum is now refused.
- ! The `session_profiling.volume_spike_ratio` config key has been removed; configs referencing it are refused.
- ! Building from source now requires Go 1.26.
- ! Contained hosts must run `sudo pipelock contain install` with the new binary; `contain upgrade` alone is not sufficient.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · This Week's Highlights · October 9, 2026
The full firehose — every feature release, every day — is in the newsletter.

▷ Subscribe