MCP SERVER | NO CODE | CLOUD HOSTED

PostHog MCP

Analyze user behavior without leaving your chat window.

PostHog lets your AI agent manage product analytics and feature flags entirely through natural conversation. Instead of jumping between dashboards to check user activity, audit flag rollouts, or review event payloads, you get a single source of truth for understanding how users behave within your app. It connects deep behavioral data analysis directly into your workflow.

A+ Quality Score 100/100

product-analytics | feature-flags | session-replay | experimentation | user-cohorts | event-tracking

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/posthog-alternative

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

PostHog MCP

13 tools available
Cloud-hosted on Vinkius

Connecting your PostHog account gives your agent full control over product analytics, feature flags, and user cohorts without ever opening the main dashboard. You can talk to it about what's happening in your application—from tracking specific user journeys to auditing which features are rolling out where. Want to know if a recent deployment changed conversion rates? Just ask. Need to see why a particular group of users isn't adopting a new feature? The agent finds the cohort and shows you their activity timeline. You can check individual user profiles, browse recent events by type, or even create historical annotations that link metric changes directly to product launches. Because this MCP is hosted on Vinkius, your agent connects once and instantly gains access to all these deep behavioral insights, making it feel like having a dedicated, expert data engineer sitting right next to you.

Core Capabilities

01 — Audit feature rollout status

Check the details of any existing or proposed feature flag, including its current enabled state and targeted user percentage.

02 — Identify specific user activity

Look up a person by their unique ID to see all their properties, last login time, and full event history.

03 — Analyze behavioral groups

List or review the definitions of dynamic user cohorts based on specific events or property filters.

04 — Track app events in real-time

Browse recent application events, filtering by event name and inspecting all associated properties for debugging or analysis.

05 — Mark key product milestones

Create annotations on the timeline to correlate specific metric changes with major deployments, launches, or incidents.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/posthog-alternative** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your PostHog Personal API Key.

02 Connect your AI client (like Cursor or Claude) to the Vinkius catalog. The agent now has access to all the analytics tools.

03 Ask a natural language question, such as 'Show me all feature flags that are currently set to 50% rollout' and let the agent execute the necessary calls.

The bottom line is you stop switching between dashboards; you just ask your AI client questions about your product data.

Built For

Product Managers who get frustrated auditing feature flag rollouts manually, or developers debugging complex user flows by sifting through raw event logs. If you spend too much time jumping between PostHog tabs just to answer a single product question, this is for you.

Product Manager

Auditing which feature flags are live and checking if the rollout percentage matches your launch plan. They use it to create annotations when a major feature ships.

Developer

Debugging complex user flows by inspecting event payloads or verifying person properties for a specific bug report without opening the PostHog UI.

Data Analyst

Reviewing behavioral cohorts and checking recent events to build reports on user engagement metrics quickly.

What Changes When You Connect

01 Stop dashboard hopping. Instead of checking the UI for `list_feature_flags` status, you just ask your agent if a specific flag is enabled and what its current rollout percentage is.

02 Deep dive into users instantly. You can use `get_person` to look up any user ID and see their entire property list and activity timeline in one go, skipping manual profile navigation.

03 Build better groups faster. Rather than manually defining filters, you can ask the agent to review all behavioral cohorts using `list_cohorts` and understand if your segmentation is accurate.

04 Debugging is easier. When an event goes wrong, you don't need to browse; you simply tell the agent to `list_events` and filter by type or time to find the faulty payload properties.

05 Contextual reporting. You can use `create_annotation` right from your chat when a major release happens, ensuring that all future metric changes are automatically linked back to that specific product launch date.

Real-World Applications

A Product Manager needs to audit a new feature.

The PM asks the agent: 'Show me all flags and which ones are running at 50% rollout.' The agent uses `list_feature_flags` and then checks the specific status of the desired flag using `get_feature_flag`, giving immediate confirmation without opening a single tab.

A Developer is debugging an intermittent bug.

The developer tells the agent: 'Find all events from user X that occurred in the last hour, specifically looking for failed purchase attempts.' The agent uses `get_person` to confirm the ID, and then runs `list_events`, immediately isolating the problematic event payload.

A Data Analyst needs to prove a marketing campaign worked.

The analyst asks: 'List all cohorts created in Q1.' The agent uses `list_cohorts` to show existing groupings, then helps review the definition of a key cohort by checking its underlying filters for accuracy.

Patterns to Avoid

Checking analytics via manual UI clicks

✗ AVOID

Opening PostHog in a browser, navigating to 'Feature Flags', clicking through dozens of flags, and then opening the 'Events' tab just to cross-reference data points.

✓ INSTEAD

Instead, ask your agent: 'What are all currently active feature flags that target users with premium status?' The agent uses `list_feature_flags` and filters based on properties in a single conversational query.

Relying on exported CSV reports

✗ AVOID

Exporting event data to a spreadsheet, spending time cleaning up timestamps, or manually matching user IDs across different tabs.

✓ INSTEAD

Ask the agent to `list_events` and filter by timestamp range. The data comes structured immediately in your chat window, ready for analysis without cleanup.

Ignoring historical context

✗ AVOID

Seeing a sudden dip in conversion rates but having no record of *why* it happened (was it a bug? A deployment?).

✓ INSTEAD

Use `list_annotations` and ask the agent to show all annotations for the last two weeks. This instantly correlates performance dips with known deployments or launches.

The Right Fit

Use this MCP if your primary job involves deep, multi-step analysis of user behavior: auditing feature flags, segmenting users into cohorts, and tracing event lifecycles to understand *why* a metric changed. It's perfect for Product Managers and Data Analysts who need real-time insight without context switching.

Don't use this if all you need is simple data retrieval, like 'What was

the total number of signups yesterday?' For that, a simpler reporting tool might suffice. This MCP excels at complex relationship mapping (e.g., linking an event to a cohort definition and then marking it with an annotation). If your goal is just single-point data collection, this might be overkill.

The Headache of Dashboard Hopping

Today, figuring out why user engagement dipped requires a painful routine. You're in the analytics dashboard checking event volume. Then you realize you need to know which feature flag controls that flow, so you have to switch tabs and manually filter by key. Next, you check the individual user profiles for examples, which means copying IDs and pasting them into another tool just to get their full activity timeline.

With this MCP, all of that manual clicking disappears. Your agent reads your request —'Check the new checkout flow.' It automatically orchestrates calls like `list_feature_flags` and then pulls relevant event data using `list_events`, giving you a unified answer in plain language. You get insights, not dashboards.

Conversational Control of PostHog

You no longer have to manually initiate the audit process. Instead of navigating through project lists and then running a separate query for cohort definitions, you ask your agent: 'Show me all currently defined cohorts that are based on purchase events.' The tool handles the sequence: finding projects via `list_projects`, listing cohorts with `list_cohorts`, and filtering them down to the relevant criteria.

The difference is control. You speak to your product data as if it were a conversation, and you get structured, actionable intelligence back immediately. It's instant depth.

PostHog Alternative: 13 Tools

These tools give you granular control over every aspect of your product data, allowing your agent to manage everything from feature flags to user annotations.

#	TOOL	DESCRIPTION
01	`create_annotation`	Marks a specific date or event (like a deployment) on the timeline for easy correlation with metric changes.
02	`create_feature_flag`	Sets up a new feature flag in your project, allowing you to control its rollout percentage and configuration.
03	`delete_feature_flag`	Irreversibly removes an existing feature flag from the system after confirming all associated data is gone.
04	`get_feature_flag`	Retrieves detailed information about a specific feature flag using its numeric ID.
05	`get_person`	Fetches all details and properties for an individual user when provided with their unique identifier.
06	`get_user`	Checks your current API key by returning the basic profile information of the connected account owner.
07	`list_annotations`	Retrieves a list of all historical annotations, showing when and what important events were marked on the timeline.
08	`list_cohorts`	Shows you all defined behavioral cohorts in your account, including their names and how they are filtered.
09	`list_events`	Gets a list of recent tracked events, letting you filter by type or check the properties attached to each event.
10	`list_feature_flags`	Lists every feature flag in your project, helping you audit which flags are active and what their configuration is.
11	`list_persons`	Retrieves a paginated list of all users tracked in the system, including their distinct IDs and last activity times.
12	`list_projects`	Displays a list of all separate analytics workspaces or projects within your PostHog account.
13	`update_feature_flag`	Modifies an existing feature flag's details, such as changing its name, description, or enabled status.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U Show me all feature flags and which ones are enabled.

I found 8 feature flags. 5 are currently enabled: 'new-checkout' (80% rollout), 'dark-mode' (100%), 'ai-suggestions' (25%), 'beta-dashboard' (50%) and 'export-v2' (100%). 3 are disabled: 'legacy-search', 'old-onboarding' and 'deprecated-api'.

U Create an annotation for today's deployment of version 3.2.0.

Done! I've created an annotation 'Deployed v3.2.0' pinned to today's date. It will appear on all relevant insights graphs so you can correlate any metric changes with this deployment.

U Show me the profile of user 'user_12345'.

Here's the profile for user_12345: email is jane@example.com, signed up 3 months ago, last active 2 hours ago. Properties include plan:premium, company:Acme Inc, and role:admin. They've triggered 847 events including 120 pageviews and 45 purchases.

Frequently Asked Questions

01 **How do I check my permissions using PostHog MCP?**

You run the `get_user` tool. This simply returns details about your connected account, confirming that your API key is valid and showing what access level you currently have.

02 **Can I list all available analytics workspaces with PostHog MCP?**

Yes, use the `list_projects` tool. This will show every separate project workspace tied to your main PostHog account ID.

03 **What is the best way to review user activity using PostHog MCP?**

The most direct way is to use `get_person`. Provide the distinct ID, and the agent will return that individual's full property set and comprehensive event history.

04 **Does PostHog MCP help me manage feature flags?**

Absolutely. You can list all existing features with `list_feature_flags`, create new ones using `create_feature_flag`, or update status via `update_feature_flag`—all conversationally.

05 **How do I link product launches to metric changes?**

You use the `create_annotation` tool. This allows you to pin a specific event, like 'V4 Launch,' to a date marker so that any future trend analysis automatically links back to it.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"posthog-alternative": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

PostHog is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by PostHog. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	PostHog MCP
Server ID	`019d8470-5702-72c0-a53f-b394d6d2b9a5`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/posthog-alternative.