MCP SERVER | NO CODE | CLOUD HOSTED

GrowthBook MCP

Control flags and environments without touching a dashboard.

GrowthBook MCP manages feature flags and experiments directly from your AI agent. Use this connector to control environments, toggle product features on or off, and organize entire experimentation roadmaps without ever opening a dashboard. It lets you run complex flag logic using natural language prompts.

A+ Quality Score 100/100

feature-flags | a-b-testing | experimentation | product-growth | deployment-control

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/growthbook

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

GrowthBook MCP

15 tools available
Cloud-hosted on Vinkius

Need to switch a feature flag for beta testing? You don't want to navigate three separate dashboards just to turn something on in production, then check if it worked in staging. This MCP connects your GrowthBook account to any AI agent, letting you manage all your product flags and experiments via simple conversation.

It lets you organize the entire experimentation roadmap by creating projects or auditing every configured environment across your stack. Need to know what flags are active? You can pull a detailed list of features or retrieve deep metadata for specific project IDs instantly. This is how development teams maintain flow, managing infrastructure changes and flag rollouts without context switching. If your current AI client supports Vinkius, you'll get access to this full catalog right where you are working.

Core Capabilities

01 — Toggle Feature Status

Turn any feature flag on or off across specific environments with a single command.

02 — Manage Project Scope

Create, read, update, or delete entire projects to keep your product experimentation organized.

03 — Audit Environments

List all configured environments—like production and staging—to verify where flags are deployed.

04 — Create Flags

Generate new feature flags for upcoming product features directly through your agent.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your GrowthBook Secret Key.

02 Connect the service to any compatible AI client (like Cursor or Claude).

03 Ask your agent to perform an action, like 'Toggle the dark mode flag in staging'—it handles the rest.

The bottom line is that you talk to it using plain English, and it executes complex infrastructure commands for you.

Built For

Product Managers who hate clicking through dashboards; DevOps Engineers tired of manual environment audits; or Engineering Leads needing to manage feature rollouts without leaving their IDE.

Product Manager

Toggling a flag for internal testing or checking the status of an experiment without logging into a separate dashboard.

DevOps Engineer

Running audits to list all configured environments and verifying project structures via natural language queries.

Engineering Lead

Creating new feature flags or updating project metadata directly from the code editor to keep development flow uninterrupted.

What Changes When You Connect

01 Saves time by eliminating manual clicks. Instead of navigating to the GrowthBook UI, you simply ask your agent to toggle a feature flag or list all project details, keeping you in your current workflow.

02 Reduces human error dramatically. You never have to copy/paste environment names or remember which specific version needs updating; just tell your agent what you want done and it handles the context.

03 Speeds up rollouts. Need to check if a beta feature is active only on staging? Use your agent to list all environments, then ask to toggle the flag—all in one conversational flow.

04 Better project organization. Instead of guessing where related flags live, you can use the MCP to create new projects or list existing ones, keeping your entire experimentation roadmap clean and auditable.

05 Deep inspection is instant. Need to know exactly what a feature's current targeting rules are? Use the `get_feature` tool to pull detailed metadata without opening any deep-dive settings pages.

Real-World Applications

Checking Status Before a Demo

The PM needs to confirm that the 'checkout-v2' feature flag is correctly enabled in the production environment before a client demo. They prompt their agent: 'List all environments and check the status of checkout-v2.' The MCP responds immediately with confirmation, saving them 15 minutes of dashboard digging.

Emergency Feature Kill Switch

A bug is found in a newly released feature. Instead of rushing through multiple tabs to disable it, the engineer asks their agent: 'Toggle the new-checkout-flow flag to OFF in production.' The change happens instantly and audibly.

Roadmap Cleanup

The team finished testing an old feature. Instead of leaving behind clutter, the lead prompts the agent to delete both the associated project container and the specific flags using `delete_project` and `delete_feature`, keeping GrowthBook clean.

Environment Drift Detection

The DevOps team suspects one environment is misconfigured. They use the MCP to list all environments, quickly spotting that 'staging' hasn't been updated with the latest settings, prompting them to run `update_environment`.

Patterns to Avoid

Using manual API calls for simple toggles

✗ AVOID

Writing a shell script that contains dozens of lines just to check if the 'beta-user' flag is on in production. This code is brittle and hard to read.

✓ INSTEAD

Just ask your agent: 'Toggle the beta-user feature flag to ON in production.' The MCP handles the complex API interaction, keeping your script clean and readable.

Overwriting project settings accidentally

✗ AVOID

Manually changing a global setting for an entire group of features without realizing it impacts other unrelated projects.

✓ INSTEAD

Use `get_project` first to fetch the specific details and check scope. Then, use `update_project` only on the intended container.

Forgetting which environment is active

✗ AVOID

Assuming that because a feature flag works in your local test setup, it's live in production. You waste time debugging in the wrong place.

✓ INSTEAD

Always start by calling `list_environments` to get a definitive list of all deployed environments before attempting any change.

The Right Fit

Use this MCP if your core problem is managing feature flag state across multiple, distinct product environments (dev, staging, production). It's perfect for teams that need instant auditability and conversational control over infrastructure settings. Don't use it if you just need to run a simple data query (use a database connector instead). Also, don't use it if your only goal is tracking user adoption metrics—that requires an analytics tool. This MCP is specifically for *control* and *state management*; its job is telling the system what to do, not analyzing how many people did it.

The Dashboard Nightmare

Right now, rolling out a simple feature flag change requires context switching. You jump from your IDE to the GrowthBook dashboard, then maybe into an environment selector, and finally click through multiple toggle switches just to confirm it’s live in staging. This process is tedious, slows development down, and invites human error.

With this MCP, you talk to your agent instead. You simply tell it what needs changing—for example, 'Toggle the new-hero-banner flag ON for beta testers.' Your agent handles all the necessary clicks, environment checks, and API calls automatically. The result is immediate confirmation, keeping you focused on code.

Control Feature Flag State with GrowthBook MCP

The ability to `get_feature` details or use the agent to `list_environments` immediately solves the headache of 'Where is this flag supposed to be active?' You don't have to guess which dashboard is correct; you just ask your agent for the facts.

You gain conversational control over complex infrastructure. It’s not about clicking buttons anymore—it’s about issuing direct, verifiable commands that keep your release cycles fast and reliable.

GrowthBook MCP: 15 Tools for Flag Management

Use these tools to manage the full lifecycle of product features, including creating flags, controlling environments, and auditing project states.

#	TOOL	DESCRIPTION
01	`create_environment`	Builds a brand-new GrowthBook environment for testing purposes.
02	`create_feature`	Creates and registers a new feature flag in the system.
03	`create_project`	Sets up an organized container for related features and experiments.
04	`delete_environment`	Removes a specified GrowthBook environment from the system.
05	`delete_feature`	Permanently deletes an existing feature flag.
06	`delete_project`	Deletes all associated resources within a project container.
07	`get_feature`	Retrieves the current settings and metadata for one specific feature flag.
08	`get_project`	Fetches all details about a specific project using its ID.
09	`list_environments`	Gives a complete list of every environment configured (e.g., production, staging).
10	`list_features`	Lists all available feature flags and their current status.
11	`list_projects`	Shows a list of every organized project within GrowthBook.
12	`toggle_feature`	Switches the state of a feature flag (on or off) for immediate testing.
13	`update_environment`	Makes changes to an existing environment's configuration settings.
14	`update_feature`	Partially modifies the metadata or targeting rules for a feature flag.
15	`update_project`	Edits the name, description, or settings of an existing project container.

See It in Action

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

U List all feature flags for the project 'frontend-v2'.

I've retrieved the features for 'frontend-v2'. You have 3 active flags: 'new-hero-banner', 'dark-mode-toggle', and 'beta-checkout'. Would you like to see the details for any of these?

U Enable the 'dark-mode-beta' feature flag in production.

I have successfully toggled the 'dark-mode-beta' flag to ON in the production environment. The change is now live.

U Get the details and settings for project ID 'proj_123'.

Inspecting project 'proj_123'... It's named 'Mobile App Redesign' with the slug 'mobile-redesign'. It currently has 12 associated features and is configured with custom targeting settings.

Frequently Asked Questions

01 **How do I list all environments using GrowthBook MCP?**

You ask your agent to perform the 'list environments' action. It returns a complete, up-to-date catalog of every environment configured in your account (like dev and production).

02 **Can I update project settings with GrowthBook MCP?**

Yes, you can use `update_project` to edit an existing container's details. This lets you organize your experimentation roadmap by changing descriptions or name tags.

03 **Does GrowthBook MCP only work for toggling features?**

No. It handles the full lifecycle, allowing you to create flags with `create_feature`, audit them with `get_feature`, and even delete them completely using `delete_feature`.

04 **What if I need to check which features exist?**

Use the 'list features' tool. This pulls a comprehensive list of every flag you have, letting you verify existence before trying to toggle or update anything.

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 `"growthbook": { "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

GrowthBook 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 GrowthBook. 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	GrowthBook MCP
Server ID	`019e38a4-8edc-73c4-ae89-00c90c874c49`
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/growthbook.