MCP SERVER | NO CODE | CLOUD HOSTED

DevCycle MCP for AI Agents

Manage feature flags and monitor deployment environments

DevCycle MCP lets your AI agent manage complex software configurations directly through natural conversation. You can list and search feature flags, monitor deployment environments (like Staging or Production), and track variable variations without writing a single line of configuration code. It gives immediate visibility into whether a feature is active for certain users or environments.

A+ Quality Score 100/100

feature-flags | experimentation | deployment-management | targeting-rules | environment-monitoring | software-testing

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

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.

DevCycle MCP

10 tools available
Cloud-hosted on Vinkius

Managing product features often means juggling dozens of toggles and checking which environment has which version. DevCycle MCP lets your AI agent handle that complexity by connecting directly to the platform's core data. Instead of logging into multiple dashboards, you simply ask your AI client questions like, 'What is the status of the beta search feature in Production?' The system retrieves the full configuration details, showing targeting rules and variations instantly.

This capability doesn't require specialized scripting; it just requires conversation. When connected via Vinkius, your agent gains operational control, allowing you to audit environment settings or even update a flag status with simple text commands. It’s designed for engineers who need rapid, reliable answers about what code is running where.

Core Capabilities

01 — Get project overviews

Retrieves high-level details and configurations for any specific DevCycle project.

02 — List all active features

Identifies every feature flag currently marked as active across your accounts.

03 — Get detailed flag configuration

Provides the full setup, targeting rules, and status for a single chosen feature flag.

04 — View all project environments

Lists every deployment stage (e.g., Development, Staging) available for your projects.

05 — Retrieve environment SDK keys

Fetches the necessary Software Development Kit keys tailored to a specific environment and platform.

06 — Search feature flags by name or tag

Finds relevant feature flags within a project using descriptive keywords.

07 — Change flag status remotely

Updates the operational state of a feature flag, changing it between active and archived directly via chat.

One Click on Vinkius — From Prompt to Execution

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

01 Connect your AI client to this MCP using your DevCycle Management API credentials.

02 Tell your agent exactly what you need—for example, 'What are the SDK keys for Production?' or 'List all active flags in Project X.'

03 The agent sends a request to DevCycle and returns structured data: flag statuses, environment details, or required security keys.

The bottom line is you get instant, conversational access to your entire software release configuration without manual dashboard navigation.

Built For

This MCP is built for technical teams that live and breathe code deployments. It helps the Site Reliability Engineer who needs to audit environment keys during an incident, or the Product Manager who just wants to know if a feature rolled out correctly before announcing it.

Software Engineer

Quickly checking flag configurations and SDK keys directly from their IDE or chat when debugging code.

DevOps/SRE

Auditing environment settings, checking deployment statuses, and managing feature flag states during incident response or large-scale rollouts.

Product Manager

Monitoring the rollout status of new features and verifying targeting rules against launch requirements during planning cycles.

What Changes When You Connect

01 Immediate auditability: Use get_feature_flag_details to pull the exact targeting rules for any flag, saving time compared to digging through UI menus.

02 Safe deployments: Monitor project environments using list_project_environments before pushing code. Know exactly if you're working in Staging or Production.

03 Operational control: Change a feature flag's status (active/archived) instantly with update_feature_flag_status, allowing for rapid kill switches during an incident.

04 Key retrieval efficiency: Get the required SDK keys for any environment using get_environment_sdk_keys; no more manual lookups in separate consoles.

05 Comprehensive visibility: List all flags and project details via list_feature_flags and get_project_details, giving a full picture of your product's configuration state.

Real-World Applications

Debugging unexpected flag behavior in Production

A Software Engineer asks their agent: 'Why isn't the beta dashboard visible for user ID 123?' The agent uses get_feature_flag_details to pinpoint that the flag is only targeting users in a different region, solving the bug immediately.

Auditing environment readiness before launch

A DevOps engineer needs to confirm all keys are ready for the next deployment. They ask for 'all Production SDK keys' and get_environment_sdk_keys delivers a list, confirming system readiness.

Disabling a problematic feature quickly

A Product Manager sees a new rollout causing errors. They instruct the agent to use update_feature_flag_status on the faulty flag. The status flips from 'Active' to 'Archived', instantly stopping the bad behavior.

Checking project structure and variables

A new team member needs context on a legacy feature. They ask for all list_feature_variables, which provides a clean inventory of every variable defined in the current project scope.

Patterns to Avoid

Manual dashboard hopping

✗ AVOID

A team member manually switches between the feature flag dashboard, the environment settings page, and the SDK key vault just to verify a single deployment status.

✓ INSTEAD

Use list_project_environments and get_environment_sdk_keys together in one chat session. Your agent pulls all necessary keys and environments into one structured response.

Assuming flag status

✗ AVOID

A developer assumes a feature is disabled because they don't see it, but the targeting rule might be set for an obscure user segment.

✓ INSTEAD

Always ask get_feature_flag_details. This tool gives you the full truth about who sees the flag and under what conditions.

Forgetting project boundaries

✗ AVOID

Trying to list all flags without first confirming which specific devcycle_project ID is relevant, leading to incomplete or irrelevant data.

✓ INSTEAD

Always start with list_devcycle_projects and then use get_project_details to narrow the scope before attempting to search for flags.

The Right Fit

Use this MCP if your workflow requires constant auditing of feature toggles, environment keys, or deployment states. If you frequently ask questions like 'Is Feature X active in Staging?' or 'What are our Production keys?', this is essential. Don't use it if you just need to create a new project—you must do that manually first. Also, don't rely on it for writing code logic; the MCP only reads and updates status; you still write the actual implementation.

DevCycle MCP: Managing Feature Flags via AI Agents

Right now, checking feature flag configurations is a nightmare. You're clicking through multiple tabs

With this MCP, you simply ask your agent about the flag setup. The response is immediate: it pulls

—the main dashboard, then into environment settings, and sometimes you have to cross-reference the targeting rules document. It’s slow, tedious, and prone to human error when tracking dozens of flags.

all the configuration details and rules into one chat window. You get instant visibility into exactly what code path a feature takes, making complex rollouts simple.

DevCycle MCP: Auditing Development Environments with AI Agents

When setting up or troubleshooting, developers spend time manually locating the correct SDK keys for Staging versus Production. They have to switch between environment panels and key vaults just to confirm they're using the right credentials.

This MCP streamlines that process entirely. A single prompt gets all necessary deployment information—like listing project environments or fetching specific SDK keys—right where you are working. You stop clicking, and start solving.

DevCycle, 10 Tools for Feature Flag Configuration

Use these tools to retrieve project details, list active features, get environment keys, and update flag statuses—all through conversation.

#	TOOL	DESCRIPTION
01	`get_environment_sdk_keys`	Lists all necessary SDK keys for every environment within a project so you can verify connectivity.
02	`get_feature_flag_details`	Retrieves the complete configuration, including targeting rules and variations, for one specific feature flag.
03	`get_project_details`	Gets detailed information about a specific project within your DevCycle account.
04	`list_active_flags`	Identifies and lists every feature flag that is currently enabled for use in the project.
05	`list_project_environments`	Shows all available deployment environments, such as Staging or Production, for a given project.
06	`list_devcycle_projects`	Lists every DevCycle project associated with your main account.
07	`list_feature_variables`	Shows all variables defined within a specific project, ensuring technical consistency across the board.
08	`list_feature_flags`	Retrieves a list of all feature flags associated with a particular project and their current operational statuses.
09	`search_feature_flags`	Searches for feature flags within a project using keywords or partial names to narrow down results.
10	`update_feature_flag_status`	Changes the operational status of a flag—marking it as active or archiving it entirely—directly through chat.

See It in Action

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

U Show me the configuration for 'New-Search-Algorithm' flag.

Feature Flag Details: New-Search-Algorithm

- **Status:** Active
- **Variations:** Control (False), Treatment (True)
- **Targeting:** 50% of users in Production.
- **Rules:** Requires user role 'premium'. Specific User IDs: [user_901, user_902].

U What are the SDK keys for our Staging environment?

Here are the required keys for your Staging deployment:

- 📱 Mobile Key: `mob_stage_abc123`
- 💻 Client Key: `client_stage_xyz789`
- ⚙️ Server Key: `server_stage_qwe456`

Do you need me to copy the client key for your team?

U List all feature flags and environments for 'Main-App'.

Project: Main-App

- **Environments Found:** [Production, Staging, Development]
- **Active Flags:** 🟢 New-Dashboard-UI (Targeting 10%)
- **Archived Flags:** 🔴 Legacy-Search (No users affected).

You can update the status of any flag with a simple command.

Frequently Asked Questions

01 **How does DevCycle MCP help me check feature flags?**

DevCycle MCP lets you ask your agent specific questions about flags, like 'What is the status of Feature X?' It doesn't just give a yes/no answer; it gives the full configuration details, including which users or environments see it.

02 **Can DevCycle MCP help me manage my deployment keys?**

Yes. You can use this MCP to list all available project environments and retrieve the correct SDK keys for any of those stages (Staging, Production). This saves you from jumping between key management consoles.

03 **What if I need to turn a feature off immediately?**

You can use this MCP to update a flag's status directly via chat. Instead of needing an engineer to deploy emergency code, your agent flips the switch from 'Active' to 'Archived', stopping the feature instantly.

04 **Does DevCycle MCP handle multiple projects?**

Absolutely. The MCP can list all projects in your account and then drill down into specific ones. You get a centralized view of flags, environments, and variables across your entire software portfolio.

05 **Is this suitable for my CI/CD pipelines?**

While it doesn't run the pipeline itself, you can use DevCycle MCP to verify pre-deployment requirements. For example, confirming all necessary environment keys are available before a merge.

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 `"devcycle": { "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

DevCycle 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 DevCycle. 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	DevCycle MCP
Server ID	`019d7584-8f59-7115-93b0-add4a90e39ad`
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/devcycle.