MCP SERVER | NO CODE | CLOUD HOSTED

# Vercel MCP

## Control deployments and domains from natural conversation.

Manage Vercel deployments, domains, and environment variables through conversation. This MCP lets your AI client take full control of complex deployment pipelines. You can list all projects across teams, inspect build logs for errors, track domain SSL status, and even cancel running builds—all without navigating the Vercel dashboard. It puts enterprise-grade DevOps operations right into your natural workflow.

A+ Quality Score 100/100

deployment | frontend-cloud | serverless | git-integration | environment-variables | edge-network

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

# Vercel MCP

14 tools available
Cloud-hosted on Vinkius

Connecting this MCP gives your AI agent full visibility into your entire deployment stack. Instead of logging into multiple dashboards to check build statuses or hunt for credentials, you simply ask your client to do it. You can list all projects with their framework and latest status. Need to debug a failing release? Your agent fetches the runtime logs instantly. Want to update a secret key? It handles creating or deleting environment variables scoped exactly to production or preview environments. Furthermore, you'll get an accurate inventory of custom domains, checking SSL health and DNS records on demand. This integration turns your AI client into a dedicated deployment engineer that knows where everything lives, making it part of the Vinkius catalog of essential dev tools.

## Core Capabilities

**01 — Audit Deployment Status**

List all projects and track their latest deployment status, including git commit information.

**02 — Manage Environment Secrets**

Create or delete environment variables across specific target environments like production or development.

**03 — Inspect Build History**

Retrieve detailed deployment history, including status changes, URLs, and commit data for every build.

**04 — Monitor Domain Health**

List all custom domains, checking their SSL certificate status and DNS verification records.

**05 — Debug Build Failures**

Fetch detailed runtime logs for any deployment to pinpoint exactly where a build or application failed.

## One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Vercel Access Token.

02 Connect the MCP to your preferred AI client (like Cursor or Claude).

03 Tell your agent what you need—for example, 'List all projects in the marketing team' or 'Check the logs for deployment X'.

The bottom line is that your AI agent manages the complex Vercel API calls so you can focus on writing code.

## Built For

This MCP is essential for DevOps engineers who spend too much time switching between dashboards, frontend developers needing quick access to secrets, and team leads who need a single source of truth for project health.

**DevOps Engineer**

Audits deployment history across multiple projects, monitors domain SSL records, and manages environment variable security using the MCP.

**Frontend Developer**

Checks if a new build is ready or fetches specific runtime logs to fix styling issues without leaving their code editor.

**Technical Team Lead**

Reviews the overall status of multiple team projects, ensuring all custom domains are configured and active before launch.

## What Changes When You Connect

01 Stop clicking through tabs. Instead of checking the Vercel dashboard for build status, simply ask your agent to list deployments or get project details.

02 Manage credentials securely. You can use create_env_var or delete_env_var to update sensitive variables across production and development environments without manual UI work.

03 Pinpoint errors instantly. If a deployment fails, running the get_logs tool fetches the necessary runtime logs immediately for debugging.

04 Verify infrastructure health. Use list_domains and get_domain to audit custom domains, confirming SSL status and DNS records are correct before launch.

05 Maintain visibility across teams. The MCP lets you list all projects or list_teams to understand who owns what, simplifying large-scale operations.

## Real-World Applications

**Investigating a Failed Build**

A developer notices an error on staging but doesn't know why. They ask their agent to get_logs for the failed deployment ID, immediately pinpointing a missing dependency in the build logs and fixing it.

**Pre-Launch Domain Audit**

A team lead needs to confirm if two new custom domains are ready. They ask the agent to list_domains, verifying both SSL certificates are active and pointing to the correct IPs before marketing goes live.

**Secret Key Rotation**

The security team mandates a key change. A DevOps engineer uses create_env_var to update the API key in production and list_env_vars to confirm the variable exists for all target environments, completing the rotation safely.

**Project Inventory Check**

A new hire needs a full picture of the application stack. They use list_projects to get an immediate inventory, seeing every deployed app, its framework, and who owns it.

# Patterns to Avoid

### Treating Vercel like a file system

✗ AVOID

A developer tries to find the last deployment by guessing a date range or manually scrolling through pages of logs. This is slow and error-prone.

✓ INSTEAD

Use list_deployments first to get an overview, then use get_deployment with the specific ID to retrieve all necessary details instantly.

### Over-relying on local cache

✗ AVOID

An engineer assumes a variable is still set because it worked yesterday. They might miss that someone deleted the key or scoped it incorrectly.

✓ INSTEAD

Always run list_env_vars for the project ID to confirm the current status and visibility of secrets.

### Ignoring team structure

✗ AVOID

Working on a global change but only focusing on one project, causing unintended breakage in other teams' services.

✓ INSTEAD

Start by running list_teams to map out the organization first, then use the resulting IDs when managing projects or domains.

## The Right Fit

Use this MCP if your job involves coordinating multiple infrastructure elements: deployments, secrets, and domain names. If you need to audit build status, check SSL records, or manipulate environment variables across environments (staging vs. production), this is the tool for you. Don't use it if your core task is merely writing content or making simple changes that don't affect deployment configuration; those tasks are better handled by a dedicated CMS connector type. If you only need to read basic project metadata, list_projects works fine, but if you ever need to interact with the build pipeline—like canceling a deployment using cancel_deployment—you must use this MCP.

## The Dashboard Rabbit Hole

Today, managing a complex web stack means bouncing between five different tabs. You check the main dashboard for overall health; then you navigate to 'Projects' to find your code base; next, you click on 'Domains' just to verify SSL status; and finally, if something fails, you have to dig into a separate 'Logs' tab. It takes clicks, context switching, and usually, at least one cup of coffee.

With this MCP, your entire stack is accessible via natural conversation. You don't click anything. You ask for the domain status, or you request the build logs for a specific deployment ID, and your agent brings all that information back in plain text. It's not just faster; it changes what 'checking' actually means.

## Get full visibility using list_projects

Before this, figuring out which projects existed and who owned them was a manual chore. You might have to check the team settings or ask a teammate for an inventory list.

Now, asking your agent to list_projects gives you that full picture instantly. It shows the project name, its framework, and its latest deployment status—all in one go. That's how much time you actually save.

# Vercel Alternative with 15 Tools

These tools allow you to perform every critical DevOps operation—from listing projects to managing environment variables—all through conversational commands.

| # | TOOL | DESCRIPTION |
|---|---|---|
| 01 | `cancel_deployment` | Cancels a running Vercel deployment by providing its unique ID. |
| 02 | `create_env_var` | Adds a new environment variable to a specific project, allowing you to scope it to multiple target environments. |
| 03 | `delete_env_var` | Permanently removes an existing environment variable from a Vercel project. |
| 04 | `get_deployment` | Retrieves specific details for a single, identified Vercel deployment. |
| 05 | `get_domain` | Fetches the configuration and status details for a specified custom domain name. |
| 06 | `get_logs` | Retrieves runtime logs, useful for debugging issues by filtering by time or project ID. |
| 07 | `list_env_vars` | Returns a list of all environment variables available in a project, showing keys but hiding values. |
| 08 | `list_projects` | Provides an overview of all Vercel projects, listing their framework, git repository, and latest deployment status. |
| 09 | `list_teams` | Lists every team within your account, providing the necessary IDs for use in other tools. |
| 10 | `get_user` | Verifies your token connection by retrieving the authenticated Vercel user's basic profile information. |
| 11 | `get_project` | Gets detailed information for a specific project, using either its ID or name. |
| 12 | `list_aliases` | Lists all active deployment aliases (URLs) mapped to specific deployments within Vercel. |
| 13 | `list_deployments` | Generates a list of all recent projects, showing their status, URL, and git commit information. |
| 14 | `list_domains` | Lists every domain configured for the team, including its verification status and DNS records. |

# 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 latest deployments for my portfolio project.

I found 5 recent deployments for your portfolio project: the latest one (READY, deployed 2h ago from main branch), followed by a BUILDING deployment from 30m ago (PR #42), and an ERROR deployment from 3h ago that failed due to a build timeout.

**U** Add the STRIPE_SECRET_KEY env var to my production environment.

Done! I've created the `STRIPE_SECRET_KEY` environment variable scoped to the production environment for your project. The value has been securely stored and will be available on the next deployment.

**U** Check if my custom domain example.com is properly configured.

The domain example.com is VERIFIED with valid A records pointing to Vercel's IPs (76.76.21.21). SSL certificate is active and issued by Let's Encrypt. The apex domain and www subdomain are both configured correctly. No DNS misconfigurations detected.

# Frequently Asked Questions

01 **How do I check my custom domains with Vercel MCP?**

You use list_domains to get a full inventory of all configured domains. This tool shows the verification status, DNS records, and SSL certificate health for every domain associated with your team.

02 **Can I update environment variables using Vercel MCP?**

Yes, you can manage secrets directly through the agent. Use create_env_var to set a new key-value pair or delete_env_var if a variable is outdated.

03 **What tool do I use for debugging deployment issues?**

Use get_logs. This function lets you retrieve runtime logs, and you can narrow down the search by project ID or specific time stamps to find the exact line of code causing the failure.

04 **How do I see all my available Vercel projects?**

You simply run list_projects. This gives you a comprehensive list showing every single deployed application, its framework, and when it was last updated.

05 **Does Vercel MCP help me cancel builds?**

Yes, if a build is running but needs to be stopped, use cancel_deployment. You just need the specific deployment ID for this action.

06 **How do I create a Vercel Access Token?**

Log in to the [**Vercel Dashboard**](https://vercel.com/account/tokens), go to **Settings > Access Tokens**, click **Create Token**, give it a name, select your team (optional) and scope. Copy the token immediately — it won't be shown again.

07 **Can I manage environment variables via the agent?**

Yes! Use `list_env_vars` to see all variable keys (values are hidden for security), `create_env_var` to add new ones with key, value and target environments (production, preview, development), and `delete_env_var` to remove them.

08 **Can I cancel a running deployment?**

Yes! Use `cancel_deployment` with the deployment ID to stop a build that is currently in progress. The deployment status will change to CANCELED. You can find the deployment ID from `list_deployments`.

09 **How do I check my deployment history?**

Use `list_deployments` optionally filtered by teamId and projectId to see recent deployments. Each entry shows the deployment URL, status (READY, BUILDING, ERROR), framework, git commit and creation date. Use `get_deployment` with a specific deployment ID for full details including build logs.

# 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 `"vercel-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

# Vercel 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 Vercel. 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 | Vercel MCP |
| Server ID | `019d8498-6b18-72ca-a508-3fa26265eb6d` |
| 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/vercel-alternative.