

M C P S E R V E R

N O C O D E

C L O U D H O S T E D

Vercel MCP for AI Agents

Manage Continuous Deployment and Domain Hosting from Chat

Vercel MCP brings your entire deployment infrastructure into chat. Instead of juggling tabs and clicking through the Vercel web UI, you can manage complex DevOps tasks conversationally. Use this connector to list projects, check live build statuses, map out custom domains, or trigger manual deployments instantly from any AI client.

A+ Quality Score 100/100

serverless

deployment

git-integration

infrastructure-as-code

web-hosting

ci-cd



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

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

Firewall & DLP

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

Agent Activity

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

Tool Health

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

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

10 tools available

Cloud-hosted on Vinkius

This MCP lets your AI agent handle advanced deployment commands normally reserved for a dedicated terminal session. You bypass the Vercel web interface entirely and manage your entire application lifecycle through chat. Need to audit which custom domains are connected? Just ask. Want to check if a recent build failed or hit a snag? Your agent finds out, instantly. It controls everything from listing all active projects to checking specific Git environment settings. You can also trigger forced builds for hotfixes when you don't want to push new code—just give the command. If your current AI client setup is limited, remember that Vinkius hosts this MCP and allows connection across multiple compatible clients, keeping your deployment workflows centralized.

Core Capabilities

01 — List all projects

Retrieves a complete list of every project configured within your Vercel account.

03 — Get build status details

Pulls specific information regarding a single deployment execution, showing its current state and history.

05 — Trigger manual build

Initiates a new, forced Vercel compilation using specific Git repository tags or branches.

02 — Audit connected domains

Maps out high-level, custom apex domains that are actively managed by your Vercel infrastructure.

04 — List recent deployments

Provides an overview of all the most recent continuous integration/continuous delivery builds for a specific project.

06 — Manage domain mappings

Lists the specific subdomain routing aliases for a project, helping map out complex DNS setups.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe your workspace to this MCP and provide your personal Vercel API Token for authorization.
- 02 Tell your AI agent what you need—for example, 'List all my active projects' or 'Cancel the current build on project X'.
- 03 The MCP sends the necessary commands to Vercel, gets the status update, and reports the result back to your chat window.

The bottom line is that you use natural conversation instead of clicking through complex web dashboards to manage every part of your deployment pipeline.

Built For

This MCP is essential for frontend engineers and DevOps architects who spend too much time switching between the Vercel dashboard, their IDE, and their terminal. If you need granular control over deployments without leaving your coding environment, this tool saves serious time.

Frontend Engineer

Needs to run a quick manual Next.js deployment inside Cursor after fixing a tricky CSS bug, while simultaneously tracking the CI steps.

DevOps Architect

Performs rapid audits on linked domain structures and checks DNS alias health across multiple organizational projects using conversational prompts.

Site Reliability Engineer (SRE)

Manages deployment lifecycles by monitoring recent builds, identifying stalled compilations to cancel them, or fetching specific preview URLs for QA testing.

What Changes When You Connect

-
- 01 You can perform advanced DevOps commands without ever touching the Vercel web UI. Your agent handles the complexity, letting you stay focused on coding.
 - 02 Instantly audit your entire infrastructure by using `list_account_domains` to map out all custom root domains in one request.
 - 03 `list_projects` gives you a quick inventory of every application deployed, so you never lose track of an environment or project ID again. It's like having the whole dashboard summarized in chat.
 - 04 If a deployment stalls, don't waste time waiting; use `cancel_active_build` to abort stalled compilations and move on with your work.
 - 05 Need to test a hotfix? Instead of committing to GitHub, you can explicitly command a forced build using `trigger_github_deployment`, saving critical minutes.
 - 06 Use `list_project_aliases` to quickly map out subdomain routing. This is huge for QA when you need to know which environment points where.
-

Real-World Applications

QA needs a specific test link

The QA team finds the target branch staging deployment is ready but they can't find the live link in the notification emails. They ask their agent to check the build status and get the exact preview URL using `'get_deployment_details'`.

Need to clean up old projects

A developer realizes three abandoned proof-of-concept sites are cluttering the account. They use `'list_projects'` first, and then safely execute `'delete_project'` on the unwanted entries.

Fixing a production bug quickly

A critical CSS bug is found right before launch. Instead of creating a branch merge, the engineer uses their agent to manually trigger deployment (`trigger_github_deployment`) directly from the main tag for immediate testing.

Mapping domain ownership

The DevOps architect needs to ensure all internal tools are correctly routed. They use `list_account_domains` and then run `list_project_aliases` on key projects to verify every subdomain is mapped correctly.

Patterns to Avoid

Trying to manually copy IDs

X AVOID

A developer tries to find the status of a deployment by clicking through 10 pages and copying a random ID, risking an incorrect or outdated reference number.

✓ INSTEAD

Instead, just ask your agent for 'all recent builds on project X' using `list_deployments`. Then, if you need more detail, feed the resulting specific deployment ID to `get_deployment_details`.

Ignoring domain conflicts

X AVOID

A developer assumes a new subdomain will work without checking which domains are already bound or what the current routing rules are.

✓ INSTEAD

First, always ask your agent to `list_project_aliases` for that project. This shows you exactly which subdomains and how they're pointing before you make any changes.

Assuming a build started

X AVOID

The engineer hits 'deploy' and assumes it's running, but doesn't know if the process was queued or failed immediately due to syntax errors.

✓ INSTEAD

Always follow up by asking for the deployment details using `get_deployment_details` with the returned ID. This confirms the status is actually progressing.

The Right Fit

Use this MCP if your workflow requires constant, rapid interaction with Vercel's CI/CD and domain management features. If you need to trigger deployments, check statuses (`list_deployments`), or audit infrastructure (like `list_account_domains`), this is your tool. Don't use it if you only need basic web hosting—those are external services. Also, don't rely on it for writing code itself; it runs commands against code that already exists. If you're just starting

and don't know what projects you have, start with `list_projects` to get oriented.

Vercel MCP and CI/CD Deployment Management

Today, managing a live deployment means switching between the Vercel web dashboard, Git source control tabs, and your terminal. You have to click through project listings, find the specific build ID, then check its status page just to see if it failed or succeeded. It's slow, tedious copy-pasting.

With this MCP, you talk directly to your agent. You simply ask for 'the deployment status on Project X,' and the agent executes all the necessary checks (like calling `list_deployments` then `get_deployment_details`) and gives you a clean, immediate answer in chat.

Vercel MCP and Domain Hosting Auditing

Auditing domains currently involves logging into the Vercel settings page for every single project to verify what custom root domain is mapped where, making it a massive headache. You have to check multiple areas manually.

This MCP lets you ask your agent to map out all your DNS and SSL status with one command. It aggregates information from `list_account_domains` and can show the specific subdomain routes using `list_project_aliases` . Everything is centralized and actionable.

Vercel: 10 Tools for CI/CD Deployment & Domain Hosting

These tools let you list projects, trigger builds, manage domain aliases, and perform deep audits on your Vercel infrastructure directly through conversation.

#	TOOL	DESCRIPTION
01	cancel_active_build	Stops an ongoing Vercel compilation pipeline immediately.
02	trigger_github_deployment	Starts a new Vercel build from a specified GitHub repository and branch reference.
03	create_project	Sets up an entirely new Vercel project given a name and framework type.
04	list_account_domains	Displays all high-level apex domains that are managed by your account.
05	list_projects	Returns a list of every Vercel project associated with the account.
06	delete_project	Permanently removes an entire Vercel project (this action is irreversible).
07	get_deployment_details	Fetches detailed status information for a specific deployment execution ID.
08	get_project_details	Retrieves the complete configuration details for any specified Vercel project.
09	list_project_aliases	Shows specific subdomain routing mappings for a given project, helping map out DNS structure.
10	list_deployments	Lists the history and status of recent CI/CD builds for one particular project.

See It in Action

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

U List all root domains connected to my Vercel infrastructure.



I've queried the apex domains across your environment. You have 3 custom domains actively bound: `production-agent.com` , `internal-tools.io` , and `legacy-blog-marketing.net` . All three signify successful certificate validations.

U Create a manual deploy on the 'billing-service' project pulling directly from the 'main' branch on GitHub repo '341xyz'.



I've successfully triggered a queued manual compilation for 'billing-service' using your target GitHub parameters.

Vercel has acknowledged the deployment intent. The new processing ID assigned to this task is `dpl_H3vM4Q...` . You can poll `get_deployment_details` intermittently with this string or I can check again for you in 3 minutes if you'd prefer.

U Check the status of deployment 'dpl_827a' and give me its exact live preview URL if ready.



I've checked deployment `dpl_827a` . Good news – Vercel marks its `readyState` as fully `READY` . There were zero pipeline errors.

You can access your newly compiled live preview here: `[https://billing-service-pr-12-test.vercel.app](https://billing-service-pr-12-test.vercel.app)`

Frequently Asked Questions

01 How does the Vercel MCP help me manage deployments without using the web UI?

You can issue all deployment commands conversationally. Your agent handles things like triggering manual builds or canceling stalled compilations, giving you immediate status updates right in your chat window.

02 Can I use Vercel MCP to audit my custom domains?

Yes, absolutely. You can ask the connector to list all high-level apex domains and then check specific project aliases to map out exactly how every subdomain is routed across your entire setup.

03 Is Vercel MCP better than using a standard CI/CD tool?

It's a complement, not a replacement. This MCP lets you **talk** to the CI/CD system directly from chat, which is faster and less disruptive than switching tabs or running complex CLI commands.

04 What if I need to delete an old Vercel project?

The connector has a tool for this. You can tell your agent to permanently remove the project, but be aware that this action is irreversible, so double-check the name first.

05 Can I list all my active Vercel projects with the MCP?

Yes, you just ask for a project listing. The connector pulls every single active and inactive project name and gives you an instant inventory of your whole account.

06 Can I cancel a stuck Vercel deployment directly from my chat window?

Yes. If a build hangs, simply find its ID via ``list_deployments`` and tell the agent: ``Abort the deployment execution 'dpl_9B...``. The ``cancelDeploymentTool`` intercepts and kills the remote CI task instantly.

07 How do I get the Preview URL of a newly built branch?

Instruct the AI manually: ``Grab the deployment details for 'dpl_xx3``'. Using the ``getDeploymentTool``, the agent will output the associated generated live ``url`` property for you to click without going to the dashboard notifications.

08 Can I spin up an empty Vercel project programmatically through the agent before I even connect my repository?

Yes. Run the prompt: `Create a new Next.js project on Vercel named 'test-saas'`. The agent initiates `createProjectTool`, and Vercel will instantly return its new unique Project GUID, ready to be linked to code whenever you wish.

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 <code>"vercel": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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	July 2026
MCP Server	Vercel MCP
Server ID	019d761b-d26d-71b0-ace9-d1f4164a2c17
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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.