MCP SERVER | NO CODE | CLOUD HOSTED

Netlify Alternative MCP

Manage deployments, sites, and forms from chat.

Netlify Alternative is an MCP that lets your AI agent manage a complete web infrastructure. It handles site listing, deployment tracking, build triggering, environment variable auditing, and real-time form submission processing—all without touching the Netlify UI.

A+ Quality Score 100/100

web-deployment | serverless-functions | build-automation | site-management | form-processing | ci-cd

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

Netlify MCP

9 tools available
Cloud-hosted on Vinkius

Managing multiple sites can be a headache. You need to check if the latest deploy worked, audit which secrets are set correctly, or see who submitted that last contact form, all before you even open your browser. This MCP gives your AI agent instant access to every piece of data governing your web platform. Your agent acts like a dedicated DevOps engineer, letting you ask questions like, 'What was the deployment status for the staging site?' or 'Show me the 10 most recent lead form submissions.' You can manage everything from listing all active sites and checking environment variables to triggering new production builds—all through natural conversation. When you connect this MCP via Vinkius, your AI client gains a single point of control over complex web operations, letting you focus on coding instead of clicking through dashboards.

Core Capabilities

01 — Audit Site Deployments

Review the full deployment history and technical details for any site or build.

02 — Manage Site Configurations

List all active sites in your account and verify current environment variables across projects.

03 — Automate Builds

Manually trigger new production builds using pre-configured build hooks.

04 — Monitor User Leads

Retrieve lists of active forms and audit recent submissions directly from your users.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Netlify Personal Access Token.

02 Your AI client connects the credentials, giving it read/write access to your web accounts.

03 You prompt your agent in natural language (e.g., 'Check the latest build for my marketing site').

The bottom line is you get a single chat window that controls all of your complex deployment and site data.

Built For

This MCP targets web developers, DevOps engineers, and project managers who spend too much time switching tabs to check on build status or audit configurations. If you're tired of manually checking dashboards at 2 am just to confirm a deployment worked, this is for you.

DevOps Engineer

Automates infrastructure checks; they can list environment variables and trigger site builds without writing shell scripts.

Web Developer

Checks the full deployment history for a specific build, or lists all sites to verify production URLs during development cycles.

Project Manager

Reviews form submissions and site status updates without needing technical logins or understanding API jargon.

What Changes When You Connect

01 Instead of navigating through build pipelines manually, you can ask your agent to list deploys or get deploy details. This saves time auditing complex technical history.

02 You gain total oversight by using list_sites to see all active web properties in one place, including critical production URLs for quick reference.

03 Monitoring leads is simple: simply prompt the agent to retrieve a list of form submissions, giving you real-time access to user data without opening a dashboard.

04 When code changes require deployment, you don't have to write commands. Just ask your AI client to trigger_site_build and watch it start working.

05 Verify security settings instantly. Use list_env_vars to check if sensitive environment variables are configured correctly across all your projects.

Real-World Applications

Investigating a broken site after deployment

A developer notices the staging site is down and asks their agent, 'What was the last deploy on my marketing-landing site?' The agent runs list_deploys, identifies the faulty build ID, and then uses get_deploy_details to pinpoint exactly which commit broke the system.

Auditing a new client's website setup

A project manager needs to verify that all sites are configured correctly for launch. They ask the agent to list_sites and then use get_site_details on each one to confirm the correct production URLs and environment variable settings.

Handling a major marketing campaign spike

The team needs to track leads immediately. Instead of checking multiple form tabs, they ask the agent to list_form_submissions, getting an instant feed of all user-submitted data from their active forms.

Preparing for a major feature release

The DevOps engineer needs to ensure every system is ready. They use list_env_vars to confirm that the new API keys are set up across accounts and sites before triggering_site_build on the main branch.

Patterns to Avoid

Treating it like a simple site dashboard

✗ AVOID

The user tries to copy-paste complex build IDs or environment variable names into the prompt, resulting in an unreadable wall of text that the agent can't parse.

✓ INSTEAD

Keep your language natural. Instead of listing technical data, ask questions like 'What are the secrets for site X?' Use list_env_vars to get a clean summary first.

Confusing build logs with submissions

✗ AVOID

The user sees a notification about a successful deploy and assumes all form data is handled, ignoring that they need explicit retrieval of lead form submissions.

✓ INSTEAD

Remember to check the user-facing data. Always follow up any deployment audit by asking for list_form_submissions if you care about leads or signups.

Overlooking site dependencies

✗ AVOID

The developer assumes that updating one site's environment variable will update related sites, leading to configuration drift and broken builds.

✓ INSTEAD

Use list_env_vars first. This tool shows you exactly which variables are set where, so you know precisely what needs changing before triggering a build.

The Right Fit

Use this MCP if your job involves auditing web infrastructure, managing multiple connected websites, or automating CI/CD tasks without constant manual UI interaction. If you need to track specific user behaviors (like viewing analytics data or database query logs), this isn't the tool for you; look into dedicated logging tools instead. However, if your workflow revolves around confirming site status, checking environment variables using list_env_vars, listing sites with list_sites, or processing form submissions via list_form_submissions, then this MCP is essential.

The Constant Web Operations Checklist

Right now, keeping a web property running means jumping between tabs: checking the build status page for new deployments, navigating to the settings panel to verify environment variables, and then opening the analytics dashboard just to see how many leads came in. It's clicking through three or four different sections of a complex UI every time something changes.

With this MCP, you stop clicking and start talking. You ask your agent to 'Audit my site config.' Instantly, it aggregates data from multiple sources—checking build statuses, listing sites, and verifying secrets—and gives you one clean answer.

Netlify Alternative: Site Management

You no longer have to manually check every deployment history or list out all your active build hooks. You can simply ask the agent, 'List my sites and their deploy histories.' The MCP pulls together data from multiple tools into one conversation thread.

Your entire web infrastructure is now accessible via natural language commands. It’s not just reading status; it's controlling the flow of work—from triggering_site_build to getting site details.

Netlify Alternative: 9 Tools for Deployment & Site Management

These nine tools allow your agent to perform deep actions like listing site metadata, triggering builds, checking secrets, and retrieving user form data.

#	TOOL	DESCRIPTION
01	`get_deploy_details`	Gathers complete technical details and logs for one specific deployment build ID.
02	`get_site_details`	Fetches detailed information and specific metadata about a single site you name.
03	`list_deploys`	Provides a summary list of deployment records for a given site, showing history at a glance.
04	`list_env_vars`	Shows all current environment variables and secrets set across your accounts and sites.
05	`list_forms`	Lists every active user-facing form configured on the site.
06	`list_build_hooks`	Displays the list of automated build hooks configured for the site.
07	`list_sites`	Retrieves a list of all active sites in your Netlify account, along with basic metadata for each one.
08	`list_form_submissions`	Retrieves a list of all recent form submissions made by your users across the site.
09	`trigger_site_build`	Initiates a new production build for a specified site name or branch.

See It in Action

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

U List all sites in my Netlify account.

Retrieving your site directory... You currently have 5 active sites, including 'marketing-landing' and 'docs-portal'. Would you like to check the latest deploy status for any of them?

U Show the last 5 deploys for site 'marketing-landing'.

Fetching deploy history... The most recent deploy was successful 2 hours ago. I see 4 other production builds from yesterday. I can provide the specific deploy IDs if you need to audit them.

U Trigger a new build for site 'docs-portal' using the 'production' hook.

Initiating build for 'docs-portal'... The build request has been accepted by Netlify. I'll monitor the status for you. It should be live in a few minutes.

Frequently Asked Questions

01 **How do I use Netlify Alternative MCP to see all my sites?**

You simply ask your agent to list_sites. It pulls a full directory of all web properties tied to your account, giving you an overview right away.

02 **What is the difference between list_deploys and get_deploy_details?**

list_deploys gives you a summarized history for a site. If you need deep technical logs or specific build IDs, use get_deploy_details.

03 **Can I check my secrets with Netlify Alternative MCP?**

Yes, you can list_env_vars. This tool shows all environment variables across your sites and accounts so you don't accidentally use an old secret key in a new build.

04 **How do I get the latest form submissions using Netlify Alternative MCP?**

To retrieve user leads, run list_form_submissions. This gets all recent data submitted through your site's active forms.

05 **If I change my site name, do I have to update anything in Netlify Alternative MCP?**

No. The agent handles the connection details using your provided token and list_sites will reflect any changes automatically.

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 `"netlify-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

Netlify 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 Netlify. 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	Netlify MCP
Server ID	`019d845f-3186-7098-8fdb-4abf1f5d44b7`
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/netlify-alternative.