MCP SERVER | NO CODE | CLOUD HOSTED

Traefik Proxy MCP

Audit your entire proxy stack conversationally.

Traefik Proxy provides real-time visibility into your edge router configuration, letting you inspect every service and rule without opening a dashboard. Connect this MCP to query routers, services, middlewares, and entrypoints across HTTP, TCP, and UDP protocols using natural conversation.

A+ Quality Score 100/100

load-balancing | reverse-proxy | infrastructure-monitoring | edge-routing | network-management

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

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.

Traefik Proxy MCP

18 tools available
Cloud-hosted on Vinkius

This connector gives your agent direct access to the operational details of your Traefik Proxy instance. Instead of jumping between multiple dashboards—one for routing, another for services, and a third for middleware—you can ask your AI client exactly what's happening at your edge. You can get a high-level summary or deep-dive into raw configuration data instantly. Need to check if traffic is reaching the correct port? Ask about entrypoints. Curious how a specific request is being transformed? Check the middlewares. It’s all available through a conversation, making complex infrastructure auditing simple. This capability becomes a core part of your existing Vinkius catalog tools, turning tedious manual checks into quick queries.

Core Capabilities

01 — Audit overall proxy health

Get an immediate summary count of all active routers, services, and entrypoints currently running.

02 — Inspect HTTP routing rules

List and get details on specific routers, services, and middlewares configured for standard web traffic (HTTP).

03 — Examine non-HTTP network connections

Query the state of TCP and UDP routers, services, and middlewares used for streaming or specialized protocols.

04 — View entrypoint status

List all configured ingress points (like web or websecure) and check their current operational status.

05 — Access full configuration data

Retrieve the entire runtime configuration of Traefik for deep debugging or compliance auditing.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Traefik API URL. Ensure the proxy's API is enabled in its configuration.

02 If your API requires it, supply any necessary basic authentication credentials.

03 Start querying the system from your preferred AI client using natural language prompts.

The bottom line is you ask a question about your network infrastructure, and your agent talks to Traefik to get the specific data back to you.

Built For

This MCP is essential for SREs and DevOps Engineers who need constant visibility into dynamic microservice environments. If you spend time clicking through multiple dashboards just to validate a single routing change, this tool saves hours.

Site Reliability Engineer (SRE)

Auditing the entire proxy configuration across protocols and verifying entrypoint consistency before major deployments.

DevOps Engineer

Quickly confirming that new services are correctly discovered, assigned routers, and have the proper middlewares applied.

Backend Developer

Checking if a newly deployed service is visible to the proxy or if its necessary routing rules were missed during deployment.

What Changes When You Connect

01 Stop clicking through multiple dashboards. By using this MCP, you can ask one question and get a full summary of the system's health with the `get_overview` tool.

02 Debugging complex traffic flows is fast. Need to know if an HTTP request hits the right rules? Use `get_http_router` or check specific middlewares with `get_http_middleware`.

03 It covers more than just web traffic. You can query non-HTTP protocols (TCP/UDP) using tools like `list_tcp_routers` and `list_udp_services`, giving you a full infrastructure view.

04 Deep debugging is effortless. If the standard views aren't enough, use `get_rawdata` to pull the complete runtime configuration for auditing or compliance checks.

05 You don't have to remember tool names. Your agent handles the complexity; you just ask what you need—be it checking a specific entrypoint via `get_entrypoint` or listing all services.

Real-World Applications

Investigating dropped traffic on a custom port

A service is reporting intermittent connection drops. Instead of logging into the proxy UI, you ask your agent to run `list_tcp_routers` and `get_tcp_service`. The agent immediately points out that the specific router needed for the high-port traffic isn't defined, solving the mystery in seconds.

Validating a new API endpoint deployment

A backend developer pushes an updated microservice and needs to ensure it gets routed correctly. They ask their agent to run `list_http_services` followed by `get_http_router`. The response confirms the service is listed and that the required host rule is active.

Performing a security audit of ingress points

The SRE team needs to confirm every exposed port matches policy. They prompt their agent for `list_entrypoints`. The system responds with a list and status, allowing the engineer to validate that only ports 80 and 443 are active.

Troubleshooting middleware transformation issues

Traffic is passing through the proxy but headers are missing. You ask your agent about `get_http_middleware` for the specific router in question, which immediately reveals that the required header modification rule was never applied.

Patterns to Avoid

Jumping between dashboards

✗ AVOID

The user opens the Traefik dashboard, clicks 'Routers,' then navigates to 'Services' to cross-reference a rule, and finally checks the 'Middlewares' tab—a multi-step process that wastes time.

✓ INSTEAD

Ask your agent directly. To check router rules and services, run `list_http_routers` and `list_http_services` in one prompt. If you need details on a specific rule, use `get_http_router`.

Assuming default settings are correct

✗ AVOID

The team assumes all services automatically register correctly, only to find critical UDP traffic is being ignored because the configuration needs explicit checks.

✓ INSTEAD

Don't trust assumptions. Use `list_udp_routers` and `list_udp_services`. This forces a full inventory of non-HTTP protocols that might be misconfigured.

Overlooking raw data access

✗ AVOID

Debugging edge cases where the API response is vague, forcing manual log file inspection.

✓ INSTEAD

Bypass human interpretation entirely. Use `get_rawdata` to pull the full, literal runtime configuration object directly into your agent's context.

The Right Fit

Use this MCP if you manage a complex microservices environment where routing rules are dynamic and span multiple protocols (HTTP, TCP, UDP). It’s perfect for SRE teams needing continuous audit visibility. Don't use it if your network setup is static or simple; in those cases, the built-in dashboard might be enough. You absolutely need this if you frequently have to check service registration status, which requires checking `list_http_services` and then validating that state with a specific router lookup using `get_http_router`. If all you need to do is view basic network topology without needing read/write access or deep inspection capabilities, simpler network mapping tools might suffice. But for runtime validation, this MCP is the standard.

Manually auditing proxy configuration feels like detective work.

Today, verifying your edge routing means logging into the dashboard, navigating to the routers tab, then opening a second panel for services. If you need to check UDP traffic, you have to switch tabs entirely and run different queries. It's slow, tedious clicking that builds context switching fatigue.

With this MCP, you simply tell your agent what you need—like 'Show me all the currently active TCP connections.' The agent runs the necessary checks under the hood and hands you a clean, consolidated answer. You get actionable data instantly.

Traefik Proxy provides full audit visibility.

Manual auditing requires copy-pasting IDs and names from one section to another just to piece together the complete picture of a single request path. You lose context, and you waste time cross-referencing middleware against router rules.

This MCP eliminates that process entirely. By asking your agent for an overview or using `get_http_router` combined with `list_http_middlewares`, you get one comprehensive view showing exactly how the data is flowing—no manual linking required.

Traefik Proxy MCP: 18 Configuration Tools

These tools let you programmatically examine every component of your proxy infrastructure, from entrypoints to specific middleware rules.

#	TOOL	DESCRIPTION
01	`get_entrypoint`	Retrieves detailed information about a specific network entry point.
02	`get_http_middleware`	Fetches the configuration details for a particular HTTP middleware.
03	`get_http_router`	Gets specific information about an active HTTP router rule.
04	`get_http_service`	Retrieves details for a specified HTTP service endpoint.
05	`get_tcp_middleware`	Gets the configuration for a specific TCP middleware.
06	`get_tcp_router`	Retrieves details about a particular TCP router rule.
07	`get_tcp_service`	Gets information on a specific TCP service endpoint.
08	`list_entrypoints`	Provides an exhaustive list of all configured network entry points.
09	`list_http_middlewares`	Lists every available HTTP middleware configuration.
10	`list_http_routers`	Lists every active HTTP router rule configured in the proxy.
11	`list_http_services`	Lists all registered HTTP services endpoints.
12	`list_tcp_middlewares`	Lists every available TCP middleware configuration.
13	`list_tcp_routers`	Lists all active TCP router rules configured in the proxy.
14	`list_tcp_services`	Lists all registered TCP services endpoints.
15	`list_udp_routers`	Lists all active UDP router rules configured in the proxy.
16	`list_udp_services`	Lists all registered UDP services endpoints.
17	`get_overview`	Retrieves a summary count of routers, services, and entrypoints across the proxy.
18	`get_rawdata`	Accesses the complete, unparsed runtime configuration data for auditing purposes.

See It in Action

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

U Give me an overview of my Traefik proxy status.

I've retrieved the overview. You currently have 12 HTTP routers, 8 TCP routers, and 15 services active across 3 entrypoints (web, websecure, and admin).

U List all active HTTP routers and their rules.

Fetching HTTP routers... I found routers like 'api-rule' (Host(`api.example.com`)) and 'dashboard' (Host(`traefik.internal`)). Would you like to inspect the middlewares for any of these?

U Show me the details for the entrypoint named 'websecure'.

Inspecting 'websecure'... This entrypoint is listening on address ':443' and has HTTP/3 enabled. It is currently handling traffic for 5 routers.

Frequently Asked Questions

01 **How do I check if a specific entrypoint is working using Traefik Proxy MCP?**

Use the `get_entrypoint` tool and provide the name of the entry point you want to inspect. This gives you details on its listening addresses and current status.

02 **Can I use the Traefik Proxy MCP for non-HTTP traffic?**

Yes, this MCP handles other protocols too. You can list and check rules for TCP and UDP using tools like `list_tcp_routers` and `list_udp_services`.

03 **What is the difference between `get_http_router` and `list_http_routers` in Traefik Proxy MCP?**

The `list_http_routers` tool gives you a full inventory of every router. The `get_http_router` tool lets you drill down into specific details for one router by name.

04 **Where do I find the complete runtime configuration with Traefik Proxy MCP?**

Use the `get_rawdata` tool. This pulls the full, unparsed JSON structure of your proxy's current state, which is useful for deep debugging.

05 **Does this MCP help me audit middleware rules in Traefik Proxy?**

Yes, you can use `list_http_middlewares` to see all available middlewares. Then, you can use `get_http_middleware` to examine how a specific rule is transforming or securing traffic.

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 `"traefik-proxy": { "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

Traefik Proxy 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 Traefik Proxy. 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	Traefik Proxy MCP
Server ID	`019e38fe-2b81-7330-acde-1da60439dcab`
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/traefik-proxy.