MCP SERVER | NO CODE | CLOUD HOSTED

Traefik Hub MCP

Govern your API routes and manage Kubernetes ingress traffic flow.

Traefik Hub MCP provides advanced API management and gateway control for cloud-native environments running on Kubernetes. It lets your agent monitor traffic latency, list all internal APIs, check service health across agents, and enforce access limits by managing subscriptions directly through the Traefik SaaS platform.

A+ Quality Score 100/100

api-management | kubernetes | ingress-proxy | traffic-monitoring | gateway | observability

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

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 Hub MCP

8 tools available
Cloud-hosted on Vinkius

Managing API routes in a large Kubernetes cluster is complicated. You need to know which services are live, what they cost in terms of traffic, and who actually has permission to talk to them—all without running manual YAML updates. This MCP lets you govern that entire layer of infrastructure through your AI client. Instead of wrestling with complex configuration files, you simply ask for the data or action you need. For example, you can check if a specific agent is healthy or see exactly which APIs are published across all your namespaces. Because this connectivity lives on Vinkius, you connect once from any MCP-compatible client and get access to these deep infrastructure controls alongside thousands of other services.

Core Capabilities

01 — Monitor API performance

Review structured data that aggregates error counts and precise latencies for all incoming API calls.

02 — Check service status

Run diagnostics to evaluate the operational health of every ingress agent deployed across your cluster hubs.

03 — Govern access permissions

Approve or block external user tokens and subscriptions, immediately severing unwanted traffic flow.

04 — Map infrastructure scope

List all active service scopes, internal APIs, and deployed agents to understand the full architecture.

One Click on Vinkius — From Prompt to Execution

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

01 First, your agent obtains necessary credentials by fetching your Platform Tokens directly from the Hub configuration.

02 Next, it safely orchestrates API traffic flow against the SaaS endpoints, evaluating the current logic bounds of your network.

03 Finally, you get back comprehensive telemetry reports detailing latencies and service health matrices for auditing.

The bottom line is that you gain immediate visibility into complex routing decisions without needing to manually interact with Kubernetes resource definitions.

Built For

This MCP is built for the Ops Engineer who gets frustrated by manual CRD updates and guessing if a service endpoint is truly live. It targets Platform Admins, Infrastructure Architects, and SREs who need deep, programmatic control over API governance and traffic monitoring.

Kubernetes Operator

Uses the MCP to check agent health (`traefik_get_agent_health`) or list active deployments to verify that all necessary ingress pods are running.

Backend Platform Admin

Manages API boundaries by listing all published APIs using `traefik_list_apis` and controlling which external users can access them via `traefik_revoke_subscription`.

SRE (Site Reliability Engineer)

Audits system performance by running reports that pull specific metrics, like latency and error rates, using `traefik_get_api_metrics` to find bottlenecks.

What Changes When You Connect

01 Real-time performance checks: Stop guessing about service health. Running `traefik_get_api_metrics` gives you detailed error traces and latency reports, telling you exactly where the bottleneck is.

02 Immediate access control: If a third party misbehaves, don't wait for manual intervention. Use `traefik_revoke_subscription` to instantly cut off their API token and restore security.

03 Full visibility into scope: You can use `traefik_list_workspaces` or `traefik_list_apis` to map out every single service endpoint, making it easy for new team members to understand the entire architecture.

04 Operational verification: Never question if your deployment is fully up. Run `traefik_get_agent_health` to check all ingress agents across the cluster and confirm they are running optimally.

05 Streamlined discovery: Instead of checking multiple dashboards, use `traefik_list_active_agents` to get a single list of every deployed pod mapped to the hub, saving hours of manual investigation.

Real-World Applications

Debugging a sudden traffic spike

A user notices high error rates. They ask their agent to run `traefik_get_api_metrics` and instantly get structured telemetries showing which specific API route is failing and why, instead of sifting through raw logs.

Onboarding a new service

A team needs to expose a new microservice. They use `traefik_list_apis` to confirm the naming conventions and then work with their manager to approve access using `traefik_approve_subscription`, keeping governance centralized.

Handling departed users

A contractor leaves the company. The Platform Admin uses `traefik_revoke_subscription` immediately, ensuring zero chance of unauthorized API calls while they are still connected to the network.

Scaling a cluster audit

Before a major deployment, an operator runs `traefik_list_active_agents` and `traefik_get_agent_health`. This confirms that every single Kubernetes agent is online and ready to handle the increased load.

Patterns to Avoid

Manual YAML changes

✗ AVOID

A developer tries to fix a routing issue by manually editing multiple Ingress resource definitions across several namespaces, risking conflicts or incorrect syntax.

✓ INSTEAD

Instead, use the MCP. First, list all current scopes with `traefik_list_workspaces` to understand the existing setup, then run `traefik_get_api_metrics` to pinpoint the exact traffic issue before making any changes.

Relying on basic logs

✗ AVOID

An SRE only looks at general cluster logs and sees a cryptic error message, but can't tell if it was an authentication failure or a service crash.

✓ INSTEAD

Use `traefik_get_api_metrics`. This tool aggregates structured data that specifically separates error traces from successful requests, giving you the root cause immediately.

Guessing agent status

✗ AVOID

A team member suspects a cluster agent is down but has to SSH into multiple nodes and check their service files one by one.

✓ INSTEAD

Run `traefik_list_active_agents` combined with `traefik_get_agent_health`. This gives you a single, authoritative pass/fail report on the entire deployment status.

The Right Fit

Use this MCP if your job involves deep governance over API gateways and Kubernetes ingress traffic. You need to know who is talking to whom, how fast they are talking, and whether the infrastructure layer itself is healthy. If you are purely focused on backend business logic (e.g., calling external REST APIs), a standard function-calling library might suffice. However, if your job involves

infrastructure concerns—like checking agent health (`traefik_get_agent_health`) or auditing which resources exist across namespaces via `traefik_list_workspaces` —you need this level of cluster visibility. Don't use this if you just need to send a message; it's for routing and access control, period.

The headache of managing API traffic boundaries

Today, knowing the state of your APIs means jumping between dashboards. You check one dashboard for agent health, another to see published routes, and a third to track latency. Every time you need to verify access or scale, it’s a manual process involving multiple clicks, cross-referencing YAML files, and copying data points into a spreadsheet.

With this MCP, that multi-system headache vanishes. You simply ask your agent to perform the audit, pulling all necessary information—from service status to published APIs—into one actionable report. What you get is an immediate, comprehensive snapshot of your entire API gateway architecture.

Traefik Hub MCP: Instant API Governance and Visibility

You eliminate the need for manual checks on agent status. Running `traefik_list_active_agents` gives you a definitive count of running pods, while `traefik_get_agent_health` confirms they are actually responding to pings.

This isn't just about reading data; it’s about control. You can use the MCP to enforce policy by executing commands like `traefik_revoke_subscription`, ensuring immediate and traceable access removal without touching a single configuration file.

Traefik Hub: 8 API Management Tools

Use these eight tools to monitor performance metrics, manage user subscriptions, list all active APIs, and audit the full operational status of your Kubernetes ingress gateway.

#	TOOL	DESCRIPTION
01	`traefik_approve_subscription`	Manually accepts a bridging logic token, which grants the user access permission through the API gateway.
02	`traefik_get_agent_health`	Tests if an ingress hub is operational by checking its live status probes across the cluster.
03	`traefik_get_api_metrics`	Collects structured reports showing API latencies and detailed error traces for performance review.
04	`traefik_list_active_agents`	Finds all running Traefik Ingress deployment pods that are currently mapped onto the service hub.
05	`traefik_list_apis`	Lists every published internal and external HTTP API route managed across the gateway.
06	`traefik_list_subscriptions`	Maps all tracked outside identities that are attempting to access resources via proxy portals.
07	`traefik_list_workspaces`	Enumerate the different logical scopes or namespaces used within the Traefik Hub.
08	`traefik_revoke_subscription`	Immediately and completely disables an active API consumer token, blocking all access.

See It in Action

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

U Scan explicitly active logic bounds listing all deployed Kubernetes Traefik Agents across our namespace hubs completely.

Processing trace limits targeting agents natively (`list_active_agents`). Hub mapped boundaries extracting safely explicitly online cluster instances running optimally effectively isolating 3 pods successfully.

U Deny active third party application logic limits explicitly mapping the execution onto subscription ID 'uuid-abc-123' natively.

Triggering restriction execution limits targeting identity explicitly naturally via `revoke_subscription`. Traefik Gateway SaaS successfully completely suspended logical mapping isolating external connections efficiently natively.

U Dump explicit gateway latencies bounding logic usage limits across the deployed API instance mapping.

Routing exact native query limitations securely onto `get_api_metrics`. Execution trace exposed logical distribution bounds verifying active successful requests gracefully isolating error clusters naturally.

Frequently Asked Questions

01 **How does traefik_get_api_metrics work?**

This tool gathers structured data on API performance. It reports specific metrics like error counts and latency measurements, allowing you to pinpoint exactly where traffic is slow or failing.

02 **What can I use traefik_list_workspaces for?**

You use this tool to see all the distinct logical scopes defined in your Traefik Hub. It helps map out the different operational areas and namespaces running within the platform.

03 **Is traefik_get_agent_health better than checking Kubernetes status?**

Yes, because it evaluates the agent's *operational* health specifically for ingress traffic. It goes beyond just confirming the pod is running and checks if it’s actually ready to route traffic.

04 **Can I use traefik_revoke_subscription to block a user?**

Absolutely. This tool immediately revokes a consumer token, effectively banning an external identity from accessing your APIs through the gateway.

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-hub": { "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 Hub 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 Hub. 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 Hub MCP
Server ID	`019d7614-74d2-73ed-aa4a-fb9b7591da29`
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-hub.