MCP SERVER | NO CODE | CLOUD HOSTED

Headscale MCP

Manage your self-hosted private mesh network.

Headscale Headscale (Tailscale Alternative) MCP connects your self-hosted private mesh network to your AI agent. Manage users, nodes, and security keys directly through conversation. You can list connected machines, create new user accounts, enforce node expirations, and control network routes without touching the command line.

A+ Quality Score 100/100

vpn | mesh-network | networking | self-hosted | identity-access | infrastructure

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

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Headscale (Tailscale Alternative) MCP

18 tools available
Cloud-hosted on Vinkius

This MCP gives you full administrative power over a private Headscale network—the self-hosted alternative to Tailscale's control server. Your AI agent acts like an embedded network administrator, letting you manage your mesh network entirely through natural language prompts. Need to audit who is on the network? You can list all connected machines and pull detailed metadata for specific nodes. If a contractor leaves, you don't need SSH access; you simply tell your agent to expire that node or user account instantly. The platform lets you generate reusable or temporary pre-auth keys for onboarding new devices. It also gives you granular control over network traffic by listing and enabling or disabling specific routes. All of this infrastructure management is available through Vinkius, the central catalog where you connect your preferred AI client.

Core Capabilities

01 — Manage Users

Create, list, or delete user accounts (namespaces) to segment and organize different parts of your network.

02 — Monitor Nodes

List all connected machines, get specific node details, rename them, move them between users, or force their session expiration.

03 — Control Access Keys

Generate new API keys and pre-auth keys; list existing credentials, then expire any key or node to revoke access immediately.

04 — Adjust Network Routes

Inspect network routes across the mesh and toggle specific routes on or off to manage data flow.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Headscale API Key and Server URL.

02 Connect your AI agent (like Cursor or Claude) to the Vinkius marketplace using these credentials.

03 Tell your agent what you need—for example, 'List all users' or 'Expire node XYZ'—and let it execute the command.

The bottom line is that your AI client becomes a hands-free interface for complex network administration tasks.

Built For

This MCP is built for DevOps Engineers and Systems Administrators who are tired of logging into a controller machine just to manage basic node lifecycle events. It's essential for privacy-conscious teams needing full control over their infrastructure.

DevOps Engineer

Quickly auditing connected nodes and managing network namespaces without leaving the terminal or IDE.

Systems Administrator (Sysadmin)

Automating the entire lifecycle of VPN nodes, including key generation and user onboarding for large teams.

Network Architect

Inspecting network routes and enforcing node expirations to maintain strict security boundaries across the mesh.

What Changes When You Connect

01 Audit node status instantly. Instead of logging into the controller and running `list_nodes`, you ask your agent to provide a full inventory, getting real-time details on every connected machine.

02 Enforce security boundaries with precision. If a contractor's laptop is compromised, instead of waiting for it to time out, you can use `expire_node` via your agent to instantly revoke all access and disconnect the device.

03 Streamline user onboarding. You no longer need manual approvals for temporary devices; simply ask your agent to `create_preauth_key`, allowing new nodes to join automatically under a specific user segment.

04 Maintain network structure with ease. Using the MCP, you can `list_routes` and then tell your agent to `enable_route` or `disable_route` for quick traffic adjustments without touching any configuration files.

05 Centralized control over credentials. Need to audit which keys are active? You can use `list_api_keys` and even call `expire_api_key` immediately if a key is found to be unused or compromised.

Real-World Applications

Revoking Access After Termination

A project manager needs to terminate access for an external team. Instead of manually tracking and revoking credentials, they prompt their agent: 'Expire all nodes associated with the billing department.' The agent handles listing users, then using `expire_node` across multiple machines.

Debugging a Broken Connection

A DevOps engineer notices an important subnet is unreachable. They ask their agent to check the network flow and confirm if routes are active. The agent uses `list_routes` and confirms which route needs to be fixed using `enable_route`.

Auditing Network Segments

A security officer needs to verify that a specific machine is correctly placed under the 'production' namespace. They ask their agent to run `list_nodes`, find the device, and then use `move_node` to confirm its placement.

Setting up Temporary Access

A team needs a temporary connection for a vendor testing a new feature. The engineer uses the agent to first `create_user`, then generate limited access using `create_preauth_key`, ensuring the access is time-bound.

Patterns to Avoid

Assuming API key management is simple

✗ AVOID

Trying to manually track which temporary keys are active and when they expire by checking a spreadsheet.

✓ INSTEAD

Use your agent with the MCP. First, run `list_preauth_keys` to see what's active, then use `expire_preauth_key` to kill access immediately if necessary.

Managing nodes manually

✗ AVOID

Logging into the web UI or CLI multiple times just to rename a machine or check its user segment.

✓ INSTEAD

Use your agent's `rename_node` and `move_node` tools. You can tell it, 'Rename this node from old-dev and move it to the finance namespace,' all in one prompt.

Ignoring network segmentation needs

✗ AVOID

Allowing general access because it's easier than setting up rules for different teams.

✓ INSTEAD

First, use `create_user` and `delete_user` to define clear organizational boundaries. Then manage traffic flow using `list_routes` and toggling routes as needed.

The Right Fit

Use this MCP if your primary need is granular, programmatic control over the lifecycle of a self-hosted mesh network. You need an AI agent to function as a central point for tasks like expiring sessions (`expire_node`), managing user boundaries (`create_user`), or adjusting traffic flow (`enable_route`). Don't use this if you just need general file storage or basic messaging; those require different types of MCPs. If your issue is simply needing a list of nodes, the `list_nodes` tool solves that immediately. However, if you also need to audit who *can* delete users (a separate concern), you might need additional authentication-focused tools in addition to this one.

Managing network access used to feel like a series of terminal commands.

Today, managing node lifecycles or revoking access means jumping between dashboards and running specific CLI commands. You have to remember the exact syntax for `list_users` versus how you delete them, making routine audits slow and error-prone. It's a manual checklist of copy/paste operations.

With this MCP, all that complexity is abstracted away. Your agent handles the execution flow—you just ask it what needs to change. You get conversational control over infrastructure that used to require deep SSH knowledge.

The Headscale MCP gives you complete node and user lifecycle management.

You no longer need to remember the difference between `delete_user` (which removes a whole segment) and simply running `get_node` (which just reads data). You can tell your agent, 'Audit the access for this node, list its user, and if it's old, expire it.'

This MCP shifts network administration from rote command execution to high-level policy enforcement. It's immediate control.

Headscale (Tailscale Alternative) with 18 Tools

Use these tools to programmatically handle every aspect of your private mesh network, from user creation and node listing to route adjustments and key management.

#	TOOL	DESCRIPTION
01	`create_api_key`	Generates a brand new API key for administrative use.
02	`create_preauth_key`	Creates a reusable or temporary pre-authentication key to allow nodes to join the network.
03	`create_user`	Establishes a new administrative user segment within Headscale.
04	`delete_node`	Removes a specific machine from the entire Headscale network roster.
05	`delete_user`	Permanently deletes an administrative user account and its associated segment.
06	`disable_route`	Turns off a specific network route, stopping traffic flow on that path.
07	`enable_route`	Activates a previously disabled network route to restore data flow across the mesh.
08	`expire_api_key`	Immediately invalidates an existing API key, requiring it to be regenerated.
09	`expire_node`	Forces a connected machine's session to expire, disconnecting it from the network instantly.
10	`expire_preauth_key`	Invalidates an existing pre-authentication key before its expiration date.
11	`get_node`	Retrieves full, detailed information about a single specified node.
12	`list_api_keys`	Provides a list of all current API keys configured on the system.
13	`list_nodes`	Pulls a complete inventory of every machine currently connected to the network.
14	`list_preauth_keys`	Lists all existing pre-authentication keys and their status.
15	`list_routes`	Displays a comprehensive list of all subnet routes and exit points in the mesh.
16	`list_users`	Lists every user account segment configured within Headscale.
17	`move_node`	Transfers a node from one user namespace to another administrative user segment.

#	TOOL	DESCRIPTION
18	rename_node	Changes the visible name of a connected machine within Headscale's records.

See It in Action

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

U List all nodes currently connected to my Headscale network.

I've retrieved the node list. You have 4 active nodes: 'web-server-01' (IP: 100.64.0.1), 'db-primary' (IP: 100.64.0.2), and two others. All are currently online except for 'backup-node'.

U Create a new user named 'external-contractors' and generate a reusable pre-auth key for them.

User 'external-contractors' has been created successfully. I have also generated a reusable pre-auth key: `hs_auth_...`. This key can be used to register multiple machines under this user.

U Get full details for node ID 12 and then expire its session.

Inspecting node 12... It's a Linux machine named 'dev-laptop-old'. I have now triggered the `expire_node` action. The machine will be disconnected and will require re-authentication to join the network again.

Frequently Asked Questions

01 **How do I use the Headscale MCP to see all connected machines?**

Run `list_nodes`. This tool pulls a complete inventory of every machine currently attached to your network, giving you a real-time picture of your infrastructure.

02 **Can I instantly revoke access using the Headscale MCP?**

Yes. You can use `expire_node` on a specific device or `expire_api_key` if an administrator key is compromised, ensuring immediate disconnection.

03 **Does the Headscale MCP help with user segmentation?**

Absolutely. Use `create_user` to establish separate segments and then use `move_node` to place machines into specific, restricted namespaces.

04 **What is the difference between a pre-auth key and an API key in Headscale?**

A pre-auth key is used for initial machine registration; you manage these with `create_preauth_key` and list them using `list_preauth_keys`. An API key is for administrative access.

05 **How do I check what network routes are active in Headscale?**

You run the `list_routes` tool. This gives you a clear overview of all subnet paths and exit points that your mesh uses for 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 `"headscale-tailscale-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

Headscale (Tailscale Alternative) 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 Headscale (Tailscale Alternative). 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	Headscale (Tailscale Alternative) MCP
Server ID	`019e38a6-9ba2-7344-bfae-a989c9a9c77d`
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/headscale-tailscale-alternative.