MCP SERVER | NO CODE | CLOUD HOSTED

Tailscale MCP

Administer your zero-trust network in natural conversation.

Tailscale MCP gives your agent full administrative control over a zero-trust mesh network. List devices, adjust access rules (ACLs), manage user identities, and audit node keys—all through natural conversation. Manage your private infrastructure without leaving your chat client.

A+ Quality Score 100/100

vpn | zero-trust | mesh-network | network-security | acl | remote-access

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

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.

Tailscale MCP

13 tools available
Cloud-hosted on Vinkius

Connecting your Tailscale network to this MCP lets your AI agent act as an administrator for your entire zero-trust infrastructure. You gain complete visibility into every device connected to your tailnet, meaning you can query details about specific nodes or check the status of all registered machines instantly. Need to tighten security? Your agent handles updating complex access control policies (ACLs), allowing you to manage network permissions without ever touching a web console or writing a manual policy file. It also manages authentication keys and users, letting you automate node joining or audit who's on the network right now. You can even delete decommissioned devices securely using their unique IDs. By connecting this MCP via Vinkius, your AI client gets all these administrative tools in one place, turning tedious infrastructure management into simple conversation.

Core Capabilities

01 — Manage Node Inventory

List every connected machine or retrieve detailed information on a specific device within the tailnet.

02 — Control Network Access Rules

Fetch and update complex access control policies, defining exactly which users and devices can communicate across your network.

03 — Audit User and Key Status

List all registered users or generate, list, and delete authentication keys for automated node joining.

04 — Secure Device Lifecycle Management

Authorize new machines to join the network, update device tags for organization, or securely remove retired devices.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Tailscale API key.

02 Your AI client uses the provided credentials to connect directly to your private network's administrative layer.

03 You interact with the system using plain language prompts, and the agent executes the required command against your live infrastructure.

The bottom line is that you manage complex networking tasks conversationally, treating your AI client like a dedicated administrator terminal.

Built For

This MCP is for the security engineer who needs to audit access policies rapidly. It's for the DevOps specialist tired of context switching between CLI and dashboard views. If you manage network identity, device provisioning, or compliance rules, this tool saves hours.

DevOps Engineer

Using the MCP to quickly list nodes, update device tags for environment separation, or generate temporary auth keys needed by a CI/CD pipeline.

Security Analyst

Checking current access control policies (ACLs) and auditing user profiles or listing all connected devices to ensure zero-trust compliance after an incident.

IT Administrator

Authorizing new employee laptops to join the network, managing device cleanup by deleting old nodes, or viewing detailed user profile information.

What Changes When You Connect

01 Stop context switching. You never need to jump from the Tailscale console to a terminal or another dashboard. Your agent performs admin tasks directly within your chat interface, saving you clicks and time.

02 Enforce strict security policies on demand. Need to change who can talk to what? Use the MCP to update network access control policies (ACLs) instantly through conversational prompts.

03 Automate onboarding and offboarding. You can generate reusable auth keys or list all users, making it simple for your agent to handle identity management without manual key generation.

04 Maintain a clean inventory. Instead of manually checking logs for old machines, you can use the MCP to list tailnet devices, audit them, and securely delete decommissioned nodes by ID.

05 Get immediate device context. Need to know if 'web-server-01' is running? You get specific details on any machine using the `get_device` tool without ambiguity.

Real-World Applications

A new developer needs access.

The IT Admin asks their agent: 'I need to add Bob's laptop and make sure he can talk to the database.' The agent runs `authorize_device` for the machine, updates the ACL using `update_tailnet_acl`, and tags it correctly with `update_device_tags`. Done in three prompts.

Security audit after a breach.

The Security Analyst asks: 'Show me every user and what access they have.' The agent uses `list_users` to get the roster, then runs `get_tailnet_acl`, providing an immediate, auditable snapshot of all network permissions.

Automating CI/CD deployment.

The DevOps Engineer needs a temporary key for a test runner. They prompt: 'Create a reusable auth key for the staging environment.' The agent uses `create_auth_key`, providing the necessary ID for secure vault storage.

Cleaning up old infrastructure.

The IT Admin notices an old IP address that should be gone. They prompt: 'Delete device 14023.' The agent uses `delete_device` to ensure the machine is fully removed from the active tailnet inventory.

Patterns to Avoid

Manual CLI key management

✗ AVOID

Copying and pasting API keys into a separate terminal session, then re-entering details in the admin web console.

✓ INSTEAD

Use `create_auth_key` to generate the necessary key directly through your agent. Keep the output secure; you'll need that ID for automation.

Ambiguous device status checks

✗ AVOID

Running a generic network health check that doesn't pinpoint which specific node is failing or why access was denied.

✓ INSTEAD

Use `list_tailnet_devices` to get the full list, and then use `get_device` on the ID of concern. This gives you precise diagnostic data.

Forgetting policy dependencies

✗ AVOID

Updating one part of the ACL (like user groups) but forgetting that another service relies on a specific tag being present.

✓ INSTEAD

Always run `get_tailnet_acl` first to see all current rules. Then, use `update_device_tags` and confirm the policy update with `update_tailnet_acl`.

The Right Fit

Use this MCP if your workflow requires deep administrative control over a zero-trust mesh network (managing devices, ACLs, and users). You need to *act* as an administrator. Don't use it if you are only trying to read basic information about the network; then simple data fetching might suffice. Critically, don't try to use this for general API calls that aren't related to device or user management—this MCP is highly specialized. If your goal is simply listing users without changing any rules, `list_users` works fine. But if you need

to *change* a rule, like using `update_tailnet_acl`, then you absolutely need this structured control.

The headache of jumping between admin consoles and terminals is real.

Today, managing network access feels like a multi-tab circus. You check the web console for device status, then switch to the terminal to generate an auth key, and finally jump back to edit policy files in a separate YAML editor. Every time you switch context, you risk making a mistake or missing a dependency.

With this MCP, your agent handles all that complexity inside your chat window. You ask it to update access rules, and boom—the network is updated. Your AI acts as the single pane of glass for your entire infrastructure.

Tailscale's administrative tools through the Tailscale MCP.

Specific manual steps that vanish include checking device tags across multiple dashboards, manually generating and storing auth keys in a secure vault, and reconciling which users have which specific rights. All these actions are now invoked by simple conversation.

Your workflow isn't just faster; it's fundamentally different. You gain conversational control over an otherwise complex command-line domain.

Tailscale MCP – 13 Tools for Network Management

These tools allow your agent to perform every administrative function required for managing a complex zero-trust mesh network, from policy updates to user auditing.

#	TOOL	DESCRIPTION
01	`authorize_device`	Approves a machine joining your tailnet.
02	`create_auth_key`	Generates a new, reusable authentication key for the network.
03	`delete_auth_key`	Removes an existing authentication key from the system.
04	`delete_device`	Permanently removes a specified device from your tailnet inventory.
05	`get_tailnet_acl`	Retrieves a full copy of the current network access control policy (ACL).
06	`get_auth_key`	Fetches details about an existing authentication key.
07	`get_device`	Retrieves specific operational details for a single device on the network.
08	`get_user`	Fetches detailed profile data for a specified user account.
09	`list_auth_keys`	Lists all current authentication keys associated with the network.
10	`list_tailnet_devices`	Provides a comprehensive list of every device connected to the tailnet.
11	`list_users`	Lists all user accounts registered within the network.
12	`update_tailnet_acl`	Modifies or updates the entire set of network access control policies (ACLs).
13	`update_device_tags`	Adds, changes, or removes organizational tags from one or more devices.

See It in Action

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

U List all devices in the tailnet 'mycompany.com'.

I've retrieved the device list for mycompany.com. You have 12 active nodes, including 'web-server-01' (ID: 14023) and 'dev-laptop' (ID: 15092). Would you like details on a specific one?

U Show me the current ACL policy for my default tailnet.

Fetching the HuJSON policy... Here is your current ACL. It defines 5 groups and 12 access rules. Would you like me to help you update any specific rule?

U Create a new reusable auth key for the tailnet 'example.com'.

I've created a new reusable auth key for example.com. The key ID is 'key-abc123xyz'. Please make sure to store it securely as it won't be shown again.

Frequently Asked Questions

01 **How does the Tailscale MCP handle user identity?**

It gives your agent full access to list users and fetch detailed profile information using `list_users` and `get_user`. This lets you audit who is on the network without needing separate tools.

02 **Can I use the Tailscale MCP to modify ACL policies?**

Yes, it's designed for that. You can retrieve existing policy files using `get_tailnet_acl` and then apply changes with `update_tailnet_acl`.

03 **What if I need to remove a device entirely?**

You use the `delete_device` tool, providing the unique ID of the machine you want gone. This ensures it's securely removed from your tailnet inventory.

04 **Do I need to manage keys separately if I use the MCP?**

No. The MCP handles key management directly. You can list existing keys with `list_auth_keys` or create a new one using `create_auth_key`.

05 **Which role should use the Tailscale MCP first?**

Security teams benefit most. They need constant visibility into who is accessing what, making the audit tools like `get_tailnet_acl` indispensable for compliance checks.

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 `"tailscale": { "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

Tailscale 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 Tailscale. 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	Tailscale MCP
Server ID	`019e38f7-ab7e-701b-8573-c1edd6364cb5`
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/tailscale.