MCP SERVER | NO CODE | CLOUD HOSTED

Cloudflare Tunnel MCP for AI Agents

Manage Zero Trust network routing and private infrastructure connections

Cloudflare Tunnel MCP lets your AI agent manage Zero Trust connectivity and private network routing for Cloudflare Tunnels. You can list tunnels, create new routes, update ingress rules, and monitor connections—all through natural conversation without touching the CLI.

A+ Quality Score 98.33/100

cloudflare-tunnel | zero-trust | cloudflared | network-security | remote-access

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

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.

Cloudflare Tunnel MCP

17 tools available
Cloud-hosted on Vinkius

Managing cloud infrastructure usually means juggling a console here and a command line there. This MCP changes that. It gives your AI client direct control over Cloudflare Tunnels, letting you handle Zero Trust connectivity purely through chat. You can list every tunnel in your account to check their health status or retrieve detailed metadata for troubleshooting. Need to update an ingress rule? You can modify origin settings and routes remotely without ever typing `cf` into the terminal. It's a massive time saver for security teams auditing network paths, or DevOps engineers quickly patching routing issues during deployments. Because this MCP is part of Vinkius, you get access to hundreds of other industry-leading tools alongside your Cloudflare setup, keeping all your infrastructure management in one place.

Core Capabilities

01 — List all tunnels and connections to check their status (healthy, degraded, or down) across the entire account.

The agent returns a consolidated report showing the operational health of every tunnel in your environment.

02 — Create or modify specific tunnel routes and IP-based network paths connecting internal resources securely.

You can define new rules that direct external traffic to specific internal services via updated routing records.

03 — Update an existing Cloudflare Tunnel, including updating its configuration and secrets.

The agent applies core changes to a tunnel's properties, ensuring the connection remains robust and up-to-date.

04 — Manage connections lifecycle

You can initiate the creation of new tunnels or fully decommission old ones, maintaining strict control over your network footprint.

05 — Audit Connectors and Sessions

The tool provides a clean way to list active connectors and run cleanup actions to remove stale sessions safely.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP on Vinkius and provide your Cloudflare API Token with the necessary Tunnel permissions.

02 Next, ask your AI client a specific question, like 'List all tunnels in my production environment,' or 'Update the ingress rules for web-app'.

03 The agent executes the request against Cloudflare's infrastructure and returns actionable data, allowing you to confirm status changes or retrieve updated configurations.

The bottom line is that your AI client acts as a single pane of glass, letting you manage complex network routing tasks without needing to memorize specific CLI commands.

Built For

This MCP is critical for DevOps Engineers who get frustrated having to switch between their IDE and the terminal just to check tunnel health. It's also essential for Security Teams that need to audit network routes against Zero Trust policy in real-time, without manually running multiple commands.

DevOps Engineer

Using this MCP to quickly update ingress rules or list tunnel connections during a deployment hotfix, avoiding manual CLI interaction.

Security Analyst

Auditing active tunnels and network routes across multiple accounts to ensure zero-trust compliance before an audit deadline hits.

System Administrator

Automating the cleanup of stale connections or creating new tunnels for temporary testing environments without leaving their primary workspace.

What Changes When You Connect

01 Instead of running five different `cf` commands, you simply ask your agent to 'List all tunnels' using the `list_tunnels` tool. It gets you a consolidated view instantly.

02 You can update complex ingress rules—like setting up a new API endpoint route—by calling `put_configuration`, which handles the syntax for you.

03 Need to check if an old connection is still active? Use `cleanup_connections` and let your agent safely prune stale sessions, ensuring high availability without manual auditing.

04 The ability to retrieve full tunnel details via `get_tunnel` means deep-diving into specific tunnels' metadata without leaving the chat interface.

05 When building a new service connection, you can use `create_tunnel` and immediately follow up with `get_tunnel_token`, all in one conversation.

Real-World Applications

Auditing Compliance After an Incident

A security analyst needs to prove that no unauthorized tunnels exist. They ask the agent to 'List and filter all tunnels by status.' The agent runs `list_tunnels` and provides a clean report, ensuring compliance with Zero Trust policies.

Deploying a New Backend Service

A DevOps engineer needs to expose a new internal microservice. They instruct the agent to 'Create a tunnel for my staging environment' (`create_tunnel`), and then use `create_route` to direct traffic to it.

Fixing Broken API Access

The main website suddenly can't reach its database. The agent checks the routes by calling `get_route_by_ip`, identifies the broken path, and uses `update_route` to fix it immediately.

Cleaning Up Old Infrastructure

A team decommissioned a project last month. Instead of manually deleting resources, they ask the agent to 'Clean up all connections for Project X,' triggering `cleanup_connections` and freeing up resources safely.

Patterns to Avoid

Over-relying on CLI scripting

✗ AVOID

A user copies a massive, multi-step deployment script into their terminal session, making it hard to read or debug the network path changes.

✓ INSTEAD

Instead of running long scripts, use your AI client to manage infrastructure state. Ask the agent to `list_routes` first, then tell it exactly which routes need changing using `update_route`. This keeps the conversation focused and traceable.

Manually managing tokens

✗ AVOID

A user has to manually copy API keys and ensure they have the correct permissions for every single action, risking security leaks.

✓ INSTEAD

Let your agent handle credentials. You can ask it to `get_management_token` and use that token within the chat context. The MCP handles the secure credential flow.

Forgetting dependencies

✗ AVOID

A user tries to update a tunnel's configuration without first confirming if the required connector is still active, causing failure.

✓ INSTEAD

Always audit first. Use `list_connections` or `get_connector` before attempting any write action like `put_configuration` or `update_tunnel`. Check the status first.

The Right Fit

Use this MCP if your workflow involves routine network auditing, rapid route adjustments, or managing a large number of Zero Trust tunnels. If you frequently find yourself checking tunnel health, updating ingress rules, or listing connections across multiple environments, this is for you. Don't use it if all you need to do is read documentation; the agent can summarize that faster. Furthermore, don't try to manage resources outside of Cloudflare Tunnel using these tools—you'll need a different type of MCP designed for those specific services.

Cloudflare Tunnel and Zero Trust: Managing Network Paths with this MCP

Today, managing your network access means jumping between the Cloudflare dashboard, running `cf` commands in a terminal, and updating firewall rules manually. If you need to change an ingress rule for a staging site, it's tedious: copy the current settings, edit them locally, then paste everything back into the command line.

With this MCP, you just tell your agent what needs fixing. You ask it to 'Update the API routes for my staging environment.' The agent handles fetching the existing configuration and applying the precise changes using tools like `put_configuration`. You get immediate confirmation that the network path is correct.

Cloudflare Tunnel and Infrastructure: Streamlining Tunnel Connections with this MCP

The biggest time sink is tracking down stale or unnecessary tunnels. Developers often spin up temporary test tunnels, and then forget to delete them, leading to security bloat. Manually listing and checking the status of dozens of these connectors is a nightmare.

Now, you just ask your agent to 'Clean up all unused connections.' The MCP runs `cleanup_connections`, identifies the stale sessions, and removes them safely. You've kept your network clean without ever leaving the chat window.

Manage 17 Cloudflare Tunnels Routes for Network Security

Use these tools to query, fetch, update, or delete tunnel routes, configurations, and connections for secure network access.

#	TOOL	DESCRIPTION
01	`cleanup_connections`	Removes specified Cloudflare Tunnel connectors from your account to maintain a clean connection list.
02	`create_route`	Creates a new network route for an existing tunnel, directing specific traffic streams.
03	`create_tunnel`	Initializes and creates a brand-new Cloudflare Tunnel within your account.
04	`delete_route`	Removes an existing tunnel route when the associated network path is no longer needed.
05	`delete_tunnel`	Decommissions and deletes a Cloudflare Tunnel completely from your service.
06	`get_configuration`	Retrieves the full configuration details for any remotely-managed tunnel.
07	`get_connector`	Fetches specific details about a Cloudflare Tunnel connector instance.
08	`get_management_token`	Generates the necessary token required to manage your tunnels programmatically.
09	`get_route_by_ip`	Looks up a specific tunnel route using an associated IP address.
10	`get_tunnel_token`	Retrieves the necessary token needed to run and connect a Cloudflare Tunnel client.
11	`get_tunnel`	Pulls all specific details for one designated Cloudflare Tunnel ID.
12	`list_connections`	Returns a comprehensive list of active and inactive Cloudflare Tunnel connections.
13	`list_routes`	Lists all defined network routes across your entire set of tunnels.
14	`list_tunnels`	Provides a filtered list of all Cloudflare Tunnels, allowing you to check health status and metadata.
15	`put_configuration`	Adds or overwrites the configuration details for a specific remotely-managed tunnel.

#	TOOL	DESCRIPTION
16	update_route	Modifies an existing network route, changing its destination or ruleset.
17	update_tunnel	Updates core properties of a Cloudflare Tunnel that are not covered by simple configuration changes.

See It in Action

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

U Show me all my tunnels and tell me which ones are currently degraded or down.

Cloudflare Tunnel Status Report

- web-server-prod: Healthy (Online)
- db-bridge: Degraded (High Latency Detected)
- test-lab: Down (Connector Failed)
- internal-api: Healthy (Online)

We recommend checking the configuration for 'db-bridge' immediately. Would you like to view its current ingress rules?

U I need to expose a new monitoring dashboard on port 80, can you set up a tunnel route?

Route Creation Successful

We created the following route:

- **Destination:** `monitoring.example.com`
- **Source:** Internet (Any)
- **Target:** `http://localhost:80`
- **Status:** Active

The new tunnel is now live. Use this endpoint to check the credentials required for access.

U Can you delete a tunnel named 'old-project' and clean up its connections?

Cleanup Complete.

1. Tunnel `old-project` has been permanently deleted.
2. All associated connectors have been removed from your account.
3. The resource cleanup is finished. Your network footprint is smaller and more secure.

Frequently Asked Questions

01 **How can I use the Cloudflare Tunnel MCP to manage my Zero Trust policies?**

You can audit your entire tunnel infrastructure by listing all tunnels and connections. This allows you to verify that every active network path adheres strictly to your company's defined Zero Trust rules.

02 **Does the Cloudflare Tunnel MCP let me update ingress rules without using the command line?**

Yes, absolutely. You can tell your agent exactly which traffic should go where—for example, directing `api.example.com` to a new internal port—and it will handle updating those complex rules for you.

03 **What if I forget about temporary tunnels? Can the MCP clean them up?**

Yes. You can instruct your agent to run cleanup actions, which safely identifies and removes stale tunnel connections and unused resources from your account.

04 **Is this Cloudflare Tunnel MCP suitable for DevOps deployment tasks?**

It's ideal for DevOps workflows. Instead of multiple manual steps, you can ask the agent to create a new tunnel and immediately establish the necessary network routes needed for testing or production.

05 **I need to check if my internal resource is exposed properly. How do I use this MCP?**

You can use the MCP to list all defined tunnel routes, allowing you to verify that your specific IP addresses and resources are connected via the correct network paths.

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 `"cloudflare-tunnel": { "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

Cloudflare Tunnel 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 Cloudflare Tunnel. 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	Cloudflare Tunnel MCP
Server ID	`019e3879-92e4-70fb-a221-3cc2005dd61a`
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/cloudflare-tunnel.