MCP SERVER | NO CODE | CLOUD HOSTED

ngrok MCP

Manage all tunnels, keys, and security rules in conversation.

ngrok MCP gives you full command over your ingress infrastructure right from your AI agent. List active endpoints, check custom reserved domains, and audit security policies—all without leaving your chat window or terminal. Manage API keys and IP restrictions to secure network access and simplify complex tunneling setups.

A+ Quality Score 100/100

tunneling | ingress | api-gateway | network-security | remote-access

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

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.

ngrok MCP

7 tools available
Cloud-hosted on Vinkius

Connecting ngrok to any AI client lets you take full control of your entire ingress infrastructure using natural conversation. Instead of navigating separate dashboards, your agent handles the heavy lifting. You can ask it to list all active public URLs—whether they're temporary tunnels or advanced edge routes. Need to check if a specific IP address is restricted? Ask about the current policies and restrictions. You also get visibility into reserved domains and API keys across your account vault. It’s like getting a complete, real-time view of every tunnel and security setting without ever leaving your chat interface. This capability fits right in with Vinkius, making it easier to manage network complexity alongside all the other tools you rely on.

Core Capabilities

01 — Check active public URLs

List temporary tunnels and permanent edge routes used by your application.

02 — Audit security rules

Review current IP policies and restrictions applied to your ngrok account.

03 — View custom domain names

Retrieve a list of all reserved, custom domains associated with your applications.

04 — Manage authentication credentials

List and manage the API keys used to authenticate access to your infrastructure.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your ngrok API Key.

02 Talk naturally with your AI client, telling it what network information you need (e.g., 'Show me all active endpoints').

03 Your agent executes the necessary commands and returns a clean summary of your infrastructure status.

The bottom line is that you get full visibility into your complex tunneling setup through simple chat prompts.

Built For

This MCP is built for engineers who manage microservices, DevOps teams needing constant infrastructure oversight, and security specialists who can't rely on outdated dashboards. If you spend time clicking between a console, an API portal, and a dashboard just to check status, this is for you.

DevOps Engineer

Auditing active endpoints and security policies across multiple environments in real-time.

Backend Developer

Checking reserved domains and HTTPS edge configurations when developing locally against a staging environment.

Security Analyst

Monitoring IP restrictions and vault usage to ensure compliance and maintain secure access boundaries.

What Changes When You Connect

01 Audit endpoints and policies instantly. Instead of logging into separate consoles to check active public URLs or IP restrictions, you ask your AI client directly for the status.

02 Keep track of custom domains easily. If you need to confirm if 'api.myapp.com' is reserved for a new project, just prompt the MCP instead of digging through domain management panels.

03 Centralize credential checks. You can list all necessary API keys and check secure vaults from one place, reducing the risk of using outdated or forgotten credentials.

04 Understand edge routing configurations. This tool lets you inspect detailed HTTPS edges, which is critical for advanced microservice deployments that require specific URL handling.

05 Secure access control review. Quickly verify IP policies and restrictions to ensure your network maintains compliance before a deployment goes live.

Real-World Applications

Debugging an intermittent public connection

A developer notices a service is intermittently failing in production. They ask their agent, 'What are the active endpoints and current IP restrictions?' The MCP immediately lists all tunnels and confirms that no external IPs are currently restricted, pointing them to a firewall issue.

Preparing for an environment migration

A DevOps engineer is moving from staging to production. They prompt the agent to 'List reserved domains' to confirm all necessary custom names are secured, followed by checking 'list_api_keys' to ensure credentials haven't expired.

Onboarding a new security team member

A security analyst needs to understand the current threat surface. They ask the agent to check both 'list_ip_policies' and 'list_vaults' simultaneously, getting an immediate overview of all access rules and stored secrets.

Verifying a new service setup

A backend developer has finished building a local microservice. They ask the MCP to list HTTPS edges and check for available reserved domains, allowing them to provision a secure public URL without manual steps.

Patterns to Avoid

Checking status across multiple consoles

✗ AVOID

The developer manually opens three tabs: the ngrok dashboard, the API key management page, and the IP policy screen. They spend 15 minutes copying and pasting data to confirm nothing is wrong.

✓ INSTEAD

Connect this MCP. Ask your agent once: 'Show me all active endpoints, reserved domains, and current IP policies.' The AI client gathers all three pieces of information instantly.

Forgetting which credentials are needed

✗ AVOID

A developer tries to connect a new service but fails because they don't know if the primary API key is expired or restricted. They waste time contacting support.

✓ INSTEAD

Use the `list_api_keys` tool through your agent. It pulls the current credential status and shows you exactly what keys are active, preventing downtime.

Misconfiguring access rules

✗ AVOID

The security team applies a new IP restriction rule but forgets to verify if it impacts existing production tunnels or reserved domains.

✓ INSTEAD

Use the agent to check both `list_ip_restrictions` and `list_reserved_domains`. You confirm that your critical assets are unaffected by the new policy before activating it.

The Right Fit

Use this MCP if you need a single conversation interface to audit network infrastructure. This is for checking status, listing resources (like endpoints or domains), and verifying policies—it's about visibility and auditing. Don't use this if your goal is to actually *modify* the underlying configuration through code; while it gives you data points, direct API calls are still needed for complex creation/deletion tasks.

If your core task involves generating boilerplate infrastructure code (e.g., Terraform or CloudFormation), you might need a dedicated Infrastructure-as-Code MCP instead. But if the job is simply 'What do I have?'—then this MCP, leveraging tools like `list_endpoints` and `list_ip_policies`, gives you the immediate answers.

Keeping track of public tunnels and security policies is a nightmare.

Every time a new microservice goes live, someone has to copy URLs from one dashboard into another, verify API keys in a separate vault, and then check the IP policy console just to make sure it's pointing to the right place. It’s constant tab switching and manual comparison.

With this MCP, you simply ask your agent what needs checking. Your AI client pulls together all active endpoints, reserved domains, and security rules into one readable summary. You get a single source of truth without ever leaving the chat.

---

ngrok MCP: Control your network access from natural conversation.

The need to switch between endpoint lists, policy screens, and key management portals disappears. You don't copy; you ask.

Now, whether you're checking `list_endpoints` or auditing `list_ip_policies`, the information flows directly into your workflow via chat. It’s instant visibility, every time.

ngrok: 7 Tools for Ingress Management

These seven tools give you granular control over every aspect of your ngrok account, allowing your agent to list everything from public tunnels to IP restrictions.

#	TOOL	DESCRIPTION
01	`list_api_keys`	Retrieves a list of ngrok API keys used for authentication across the account.
02	`list_endpoints`	Gets a comprehensive list of all active public URLs (tunnels) configured in the system.
03	`list_https_edges`	Lists advanced HTTPS edge configurations and their associated routing rules.
04	`list_ip_policies`	Shows all IP policies currently configured, detailing access controls by IP range.
05	`list_ip_restrictions`	Lists specific IP restrictions applied to the account dashboard and API access.
06	`list_reserved_domains`	Retrieves all custom domains that have been reserved for future use in applications.
07	`list_vaults`	Lists the secure vaults used to store sensitive values and credentials.

See It in Action

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

U List all active ngrok endpoints.

I've retrieved your active endpoints. You have 2 active tunnels: 'https://myapp.ngrok-free.app' (ID: ep_123) and 'https://api.example.com' (ID: ep_456).

U Show me the reserved domains in my account.

Fetching reserved domains... I found 3 domains: 'dev.local', 'staging.myapp.com', and 'prod.myapp.com'. All are currently active.

U What IP policies are currently configured?

I found 1 IP policy: 'Office-Only-Access' (ID: ipp_789) which restricts access to the corporate CIDR range.

Frequently Asked Questions

01 **How do I use ngrok MCP to check my active tunnels?**

Use the `list_endpoints` tool through your agent. It pulls a real-time list of all public URLs, letting you see which tunnels are currently running and visible.

02 **Can I use ngrok MCP to check if my custom domain is reserved?**

Yes, prompt the agent about 'reserved domains.' It uses `list_reserved_domains` to show all custom names you've set aside for future projects.

03 **What does ngrok MCP use to manage credentials?**

The MCP interacts with your account vault, allowing you to list and audit API keys using the `list_api_keys` tool. This keeps sensitive values managed and visible to your agent.

04 **Is ngrok MCP only for checking endpoints?**

No. It handles security too. You can use `list_ip_policies` and `list_ip_restrictions` to verify access controls, not just the public URLs themselves.

05 **Does ngrok MCP help with HTTPS edge configurations?**

Yes. You can specifically ask about advanced routing using the `list_https_edges` tool, giving you insight into complex URL handling rules.

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 `"ngrok": { "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

ngrok 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 ngrok. 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	ngrok MCP
Server ID	`019e38c8-100d-7292-8c12-90cd5207bc74`
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/ngrok.