MCP SERVER | NO CODE | CLOUD HOSTED

Cloudflare

A+ Quality Score 98.33/100

serverless | edge-computing | infrastructure-as-code | api-management | deployment-automation | secrets-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026

vinkius.com/mcp/cloudflare

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 MCP

25 tools available
Cloud-hosted on Vinkius

One Click on Vinkius — From Prompt to Execution

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

Built For

25 Tools Available

#	TOOL	DESCRIPTION
01	create_deployment	Strategy can be immediate (100% traffic immediately) or gradual (percentage-based rollout). Requires script name, version ID, and deployment strategy. Use this to roll out new features, rollback to previous versions, or perform canary deployments. Deploy a specific Worker version to traffic
02	create_secret	Secrets are encrypted at rest and injected at runtime. Requires script name, secret name, and secret value. Common use: API keys, database passwords, OAuth tokens. The secret becomes available via env.VARIABLE_NAME in your Worker code. Create or update a secret for a Cloudflare Worker
03	create_tail_session	log() output and exceptions. Returns a tail ID and WebSocket URL for streaming logs. Use this for debugging Workers in production or monitoring error output. Create a tail logging session for a Cloudflare Worker
04	create_worker_route	Requires zone ID, URL pattern (e.g., "example.com/api/*"), and script name. Use this to expose your Worker at specific URL paths or domains. Create a new route pattern for a Cloudflare Worker
05	delete_secret	Use this to clean up unused secrets or rotate credentials. Requires script name and secret name. After deletion, the Worker will no longer have access to the secret value. Delete a secret from a Cloudflare Worker
06	delete_tail_session	Requires script name and tail ID. Use this to clean up unused tail sessions when debugging is complete. Delete a tail logging session for a Cloudflare Worker
07	delete_worker_route	Use this to stop serving a Worker at specific URLs. Requires zone ID and route ID. Delete a route pattern from a Cloudflare Worker
08	delete_worker	This action cannot be undone. Requires the script name. Confirm with the user before proceeding. Delete a Cloudflare Worker script and all its associated resources
09	get_kv_key	Returns the raw value as JSON. Use this to read configuration values, cached responses, or user data stored in KV. Get the value of a specific key in a KV namespace
10	get_worker_analytics	Returns data for recent invocations. Use this to monitor Worker performance, identify errors, or track usage trends. Get analytics data for a specific Cloudflare Worker

#	TOOL	DESCRIPTION
11	get_worker	Requires the script name from list_workers results. Use this to review Worker configuration before making updates or debugging. Get detailed information about a specific Cloudflare Worker
12	get_worker_version	Requires script name and version ID from list_worker_versions results. Use this to audit version contents or prepare for rollback deployment. Get detailed information about a specific Worker version
13	get_zone_analytics	Returns aggregated data for the last 24 hours. Use this to monitor traffic patterns, identify spikes, or measure CDN performance. Get analytics data for a specific Cloudflare zone
14	list_d1_databases	Returns database IDs, names, creation dates, and file sizes. Use this to identify available databases before querying. List all D1 databases in your Cloudflare account
15	list_deployments	Returns deployment IDs, version IDs, strategies (immediate, gradual), creation dates, and traffic percentages. Use this to review current deployment state, monitor gradual rollouts, or identify which version is live. List all deployments for a specific Cloudflare Worker
16	list_kv_keys	Returns key names, expiration metadata, and sizes. Use this to audit stored data or find specific keys before reading values. List all keys in a specific KV namespace
17	list_kv_namespaces	KV namespaces are key-value stores for Workers. Returns namespace IDs, titles, and creation dates. Use this to identify which namespaces exist before reading/writing data. List all KV namespaces in your Cloudflare account
18	list_r2_buckets	Returns bucket names, creation dates, and storage locations. Use this to identify available storage buckets before managing objects. List all R2 storage buckets in your Cloudflare account
19	list_secrets	Returns secret names and types (secret_text, secret_key). Secret values are never returned for security. Use this to audit which secrets are configured before adding new ones or cleaning up unused secrets. List all secrets for a specific Cloudflare Worker
20	list_worker_routes	Returns route patterns, associated script names, and zone IDs. Use this to understand which URLs invoke your Worker before adding or removing routes. List all route patterns associated with a Cloudflare Worker
21	list_worker_versions	Each version represents a deployed code snapshot with unique ID, creation date, and metadata. Returns version IDs, timestamps, and author information. Use this to review deployment history, rollback to previous versions, or audit code changes. List all versions of a specific Cloudflare Worker

#	TOOL	DESCRIPTION
22	list_workers	Returns script names, creation dates, modification dates, and deployment status. Use this as the first step to identify which Workers exist before managing versions, deployments, or secrets. List all Cloudflare Workers scripts in your account
23	list_zones	Returns zone IDs, domain names, status, plan, and name servers. Use this to identify zone IDs needed for Worker routes, DNS management, or cache operations. List all DNS zones in your Cloudflare account
24	purge_cache	Use this after deploying content changes or updating static assets. Requires zone ID. Purge all cached content for a specific zone
25	query_d1	Supports SELECT, INSERT, UPDATE, DELETE operations. Returns query results as JSON. Use this for data analysis, migrations, or ad-hoc queries. Requires database ID and SQL query string. Execute a SQL query against a D1 database

See It in Action

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

U List all serverless Cloudflare Workers deployed natively bound to my account.

Pulling instances via `list_workers`. Extracting exactly 5 active edge-computation scripts securely operating on your endpoints mapping perfectly.

U Query the KV namespace assigned to 'production_keys' and extract the specific text mapping 'gateway_url'.

Connecting KV namespaces natively... Resolved variable payload matching 'gateway_url': `api.vinkius.cloud` loaded efficiently without cache interference.

U Check error statistics on my main D1 SQLite database instance over the last 24 hours.

Parsing telemetry array from metrics. The D1 metrics reveal stable instances with precisely 0% crash loops inside the isolated runtime edges.

Frequently Asked Questions

01 **Can I deploy new Worker scripts directly through the AI agent?**

Yes! You can orchestrate deployments natively mapping source code variables seamlessly without touching Wrangler.

02 **Does it interact with specific D1 Serverless SQL definitions?**

Absolutely. Ask your agent to parse complex D1 databases and query tables instantly utilizing precise analytical calls.

03 How are environment variables and secrets handled?

All `add_secret` queries execute symmetrically securely masking output variables while committing seamlessly inside Cloudflare matrices.

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": { "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 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. 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 MCP
Server ID	`019d7574-1eda-704c-b226-9019c82c8dc7`
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.