MCP SERVER | NO CODE | CLOUD HOSTED

Turso MCP

Manage distributed databases with conversation.

Turso MCP lets you manage complex serverless SQLite databases and their edge locations entirely through conversation. Connect your AI agent to orchestrate database provisioning, rotate security tokens, list global infrastructure details, and clean up unused resources without opening a terminal.

A+ Quality Score 98.33/100

sqlite | edge-computing | serverless | database-orchestration | jwt-authentication | data-persistence

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

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.

Turso MCP

10 tools available
Cloud-hosted on Vinkius

This MCP gives your AI client total control over your Turso cloud infrastructure. You can interact with the system as if you were running direct CLI commands, but right within your chat window. Need to spin up a new test database? Just ask. Want to check which physical data centers host your current instances? Ask that too.

The tool lets you look up organizational tenants, map global Fly.io locations, and even provision entirely new, distributed databases on demand. Furthermore, securing the system is easy; you can list active connection tokens or revoke all old ones instantly when a key gets compromised. It's built for platform engineers who hate context-switching between their IDE and their command line. Because this MCP lives in Vinkius, you connect once from any compatible agent and get access to these core database operations alongside hundreds of other specialized services.

Core Capabilities

01 — Database Provisioning

Create entirely new distributed SQLite databases using just the organization slug, database name, and target group.

02 — Infrastructure Mapping

Identify root organizational tenants or look up physical global datacenter locations to understand your data topology.

03 — Database Status Checks

Retrieve the full list of existing databases, logical group configurations, or detailed architectural metadata for a specific instance.

04 — Security and Access Control

Generate secure connection tokens tied to specific databases, list all active JWTs, or permanently invalidate old credentials across the board.

05 — Database Cleanup

Permanently delete global libSQL database instances that are no longer needed, preventing resource leakage.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Turso API Token.

02 Your AI agent connects using the token and gains read/write access across all defined database tools.

03 You initiate a conversation asking for an infrastructure action, like 'list all my databases' or 'create a new test db'.

The bottom line is you manage your entire distributed database lifecycle through natural language prompts from any compatible client.

Built For

Platform engineers and DevOps teams need this. If you spend time context-switching between a dashboard, documentation, and the CLI to manage databases, this MCP saves that mental overhead.

DevOps Engineer

You run migrations or validate token expirations by asking your agent directly instead of scripting complex bash commands.

Platform Architect

You map out global database fleets, check which physical data centers are active, and identify zombie resources for cleanup.

Backend Developer

You provision new test environments on the fly or verify libSQL metadata without leaving your IDE workflow.

What Changes When You Connect

01 Stop context switching. You can provision a new test database and check its metadata without ever leaving your IDE or chat interface. This is pure, conversational infra-ops.

02 Maintain robust security by instantly rotating credentials. Instead of manually finding old tokens to delete, you simply ask the agent to rotate them for any instance.

03 Get full visibility into your infrastructure. Need to know which data center is hosting a specific database? Use the tool to map global locations and understand your physical reach.

04 Control resource sprawl with precision. Identify and permanently remove 'zombie databases' or unused resources that cost money but nobody tracks.

05 Manage groups and organizations easily. You can list all logical group configurations, which helps you scope out where a new database needs to live.

Real-World Applications

Auditing Access Tokens

You suspect an old employee left behind credentials for your production DB. Instead of manually checking every service config, you ask the agent to list all active database tokens and then use 'rotate_database_tokens' on the specific instance.

Spinning up a Test Environment

A developer needs a dedicated staging DB for feature testing. They prompt the agent, which uses `create_database` to provision the environment and then immediately prompts them if they want a connection token via `create_database_token`.

Global Topology Check

The platform team needs to confirm that their data is backed by multiple physical regions. They ask the agent to run through `list_edge_locations`, giving them an instant map of available global fly.io datacenter mappings.

Cleaning up Old Projects

A project was shelved, leaving behind a massive database instance consuming resources. You use the agent to list all databases first (`list_databases`), then confirm and run `delete_database` on the target name.

Patterns to Avoid

Running multiple CLI commands

✗ AVOID

The user opens their terminal, runs 'turso list-dbs', copies a hostname, then switches to another tab and manually types 'turso get-details --host=<hostname>' for every single database.

✓ INSTEAD

Instead, you ask your agent: 'List all my databases and give me the architectural details for each one.' The agent handles the entire sequence using `list_databases` and then calling `get_database_details` automatically.

Assuming token status

✗ AVOID

A developer thinks a connection is broken because they can't connect, so they just assume the old token expired without confirmation.

✓ INSTEAD

First, use `list_database_tokens` to verify which tokens are active and if any need rotation. If necessary, you run `rotate_database_tokens` immediately.

Forgetting about groups

✗ AVOID

A developer tries to create a database but doesn't know which logical group it should belong to, resulting in an error and manual investigation into the schema.

✓ INSTEAD

Before creating, you run `list_database_groups` to see all available organizational containers. Then you use that name when calling `create_database`.

The Right Fit

Use this MCP if your core job involves managing infrastructure state: provisioning, deleting, and securing distributed databases across multiple locations. If you need to know the current topology or validate connection credentials, this is your tool.

Don't use it if you are simply trying to write data (e.g., running CRUD operations). For that, you'll need a direct client library connection. Also, don't rely on it for high-frequency runtime metrics; use dedicated monitoring tools for those performance numbers. This MCP is about orchestration and governance, not live traffic metrics.

Dealing with fragmented database control panels is exhausting.

Right now, managing your distributed databases means jumping between three different places: the cloud provider's web portal to see global locations; a separate CLI tool to list existing instances and their metadata; and finally, a secrets manager to track active connection tokens. You spend more time gathering information than actually coding.

With this MCP, all that data lives in one place. Your AI agent acts as the central control plane, letting you ask complex questions like 'What is my current database footprint?' or 'Show me all unrotated credentials,' and getting a clean answer without ever touching a dashboard.

The Turso MCP gives you full Database Lifecycle Operations.

You eliminate the manual steps of listing group configurations, then checking location maps, and finally running a delete command on three separate screens. You tell your agent, 'Clean up all test databases in the staging environment,' and it handles the entire multi-step process.

The difference is that database management stops being a series of disjointed clicks or commands; it becomes a single conversation.

Turso: 10 Database & Location Tools

These tools let you monitor, track, profile database locations, groups, tokens, and execute core lifecycle management actions against your Turso cloud infrastructure.

#	TOOL	DESCRIPTION
01	`create_database`	Provisions a massively distributed Serverless SQLite database for use in your application.
02	`create_database_token`	Mints a secure, temporary connection token that is tied only to a specific database instance.
03	`delete_database`	Permanently deletes an entire global libSQL database; this action cannot be reversed.
04	`get_database_details`	Introspects and returns the exact architectural traits of a specific target libSQL instance.
05	`rotate_database_tokens`	Revokes all previously issued connection tokens for a given database, forcing clients to use new credentials.
06	`list_databases`	Generates an enumeration of the entire libSQL Edge Database registry in your organization.
07	`list_database_groups`	Retrieves a list of logical groups that organize and manage database locations.
08	`list_edge_locations`	Looks up the physical global datacenter mappings, showing where your data can be hosted.
09	`list_organizations`	Identifies and lists all root organizational tenants within Turso Edge SQLite.
10	`list_database_tokens`	Lists all currently active JWT tokens used for database execution, allowing you to audit access.

See It in Action

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

U List all my Turso databases and their hostnames.

You have 3 databases in the organization 'startup-inc': 1. 'prod-db' (prod-db-startup.turso.io), 2. 'staging-auth' (staging-auth-startup.turso.io), and 3. 'analytics-edge' (analytics-edge-startup.turso.io). Do you need details or connection tokens for any of them?

U Create a new database named 'user-cache' in the 'default' group.

I've successfully provisioned the 'user-cache' database. It has been allocated to the 'default' group and is available at 'user-cache-startup.turso.io'. Would you like me to mint a connection token for it?

U Invalidate all tokens for the 'legacy-db' database immediately.

All previous JWT connection tokens for the 'legacy-db' instance have been successfully rotated and invalidated. Any clients using the old credentials will now be denied access. Let me know if you need to generate a new replacement token.

Frequently Asked Questions

01 **How do I list all my databases using Turso MCP?**

You ask the agent to execute 'list_databases'. This provides you with the full registry of every libSQL Edge Database instance currently managed under your account.

02 **Can I use Turso MCP to delete a database?**

Yes, you can. Use the 'delete_database' tool when you want to permanently remove an entire global libSQL database. Be careful because this action is irreversible.

03 **What if my connection token expires? How do I fix it with Turso MCP?**

You can first check which tokens are active using 'list_database_tokens'. If they're stale, you run 'rotate_database_tokens' to immediately revoke all old credentials.

04 **Does the Turso MCP help me find global data center locations?**

Yes. You can use 'list_edge_locations' to look up physical global datacenter mappings, giving you a clear view of where your database infrastructure operates.

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 `"turso": { "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

Turso 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 Turso. 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	Turso MCP
Server ID	`019d7616-8ebc-71a4-bb7b-5c9d9d91977b`
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/turso.