MCP SERVER | NO CODE | CLOUD HOSTED

Axiom MCP for AI Agents

Querying and Managing Cloud Observability Logs

Axiom manages observability and log data directly through your AI client. You can ingest raw logs (JSON, CSV) into structured datasets, run powerful Axiom Processing Language (APL) queries in real-time, and manage complex infrastructure components like monitors and dashboards using natural conversation.

F Quality Score 3.6/100

telemetry | log-analysis | real-time-monitoring | data-ingestion | apl-queries | cloud-observability

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

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.

Axiom MCP

31 tools available
Cloud-hosted on Vinkius

Connecting your Axiom account to any AI agent lets you handle log management and observability tasks right where you work. Instead of jumping between a terminal, a database UI, and a monitoring dashboard just to answer one question, you talk to your AI client. It handles the heavy lifting: ingesting massive amounts of raw data into structured datasets or running complex APL queries against live logs. You can manage everything from creating new monitors for alert checks to listing user details needed for auditing, all through simple prompts. If your current workflow involves manual data cleanup and stitching together information from separate monitoring tools, this MCP changes that. Vinkius hosts the Axiom connection, allowing you to access these powerful data controls instantly from any compatible client.

Core Capabilities

01 — Running advanced log queries

Execute complex APL queries against your ingested datasets to count errors, identify trends, or filter logs based on specific criteria.

02 — Managing telemetry components

Create and delete critical infrastructure elements like monitors, dashboards, and notifiers that track system health and trigger alerts when thresholds are breached.

03 — Organizing raw data streams

Ingest various file types (JSON, CSV) into managed datasets or list existing ones to keep your infrastructure telemetry organized and ready for querying.

04 — Auditing user access details

Retrieve information about users, API tokens, and organization settings needed for security audits and access control management.

05 — Visualizing system performance

Create new dashboards or retrieve existing ones to visualize trends, track key metrics over time, and annotate significant events on the timeline.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and provide your Axiom API Token along with any required Organization ID.

02 Next, connect your preferred AI client (like Cursor or Claude) to the Vinkius catalog. The connection validates your credentials.

03 Finally, you simply ask your agent what you need—for instance, 'Show me all monitors checking latency'—and it executes the necessary commands using Axiom.

The bottom line is: you talk to your AI client once; it handles the complex data operations with Axiom for you.

Built For

This MCP targets technical roles that live in a constant state of firefighting. It's for the DevOps engineer who is tired of switching between five different tabs just to check if a service broke, or the Data Analyst struggling to query large, unstructured log dumps. If your job requires understanding what went wrong at 3 AM using raw system data, this MCP belongs on your stack.

Site Reliability Engineer (SRE)

Runs immediate health checks, querying logs for specific error codes or updating monitors to track new performance metrics.

DevOps Engineer

Manages the lifecycle of telemetry data by creating datasets and managing alert notifiers without leaving their primary chat interface.

Software Engineer

Debugs production issues instantly, running APL queries against recent logs to find specific trace IDs or user-related error messages.

Data Analyst

Ingests and analyzes massive, unstructured log dumps by defining datasets and using the processing language to extract meaningful metrics.

What Changes When You Connect

01 Instead of manually checking logs, you tell your agent to run a query. It executes the complex APL command using `run_query` and delivers the results directly in the chat.

02 You gain full oversight by managing all visibility components—from setting up new alerts with `create_monitor` to updating notification rules via `update_notifier` —all through conversation.

03 Stop wasting time organizing data. You can ingest raw, messy files using `ingest_data` and immediately structure them into clean datasets ready for analysis.

04 Auditing is faster: Use `list_users` , `get_user` , or `list_tokens` to instantly gather the access details needed for compliance checks without navigating complex admin portals.

05 Visualizing trends gets easier. You can create new dashboards (`create_dashboard`) and annotate important events using `create_annotation` , keeping all context in one place.

Real-World Applications

Investigating an unexpected traffic spike

A user asks, 'What caused the latency dip last night?' The agent runs a targeted APL query against the production logs and returns a table showing the correlation between increased API calls and error rates.

Setting up compliance monitoring for PII

A user commands, 'Create a monitor that alerts if any dataset contains unmasked PII.' The agent runs `create_monitor` and sets up the required notification rule to prevent leaks.

Onboarding a new team member's access

A user asks, 'What tokens does Jane Doe have?' The agent retrieves her profile using `get_user` and then lists all active API tokens associated with her account for review.

Debugging an intermittent production bug

The engineer prompts, 'Show me the logs related to trace ID XYZ.' The agent executes a query and presents the relevant log snippets and user details, immediately narrowing down the scope of the issue.

Patterns to Avoid

Over-relying on manual UI clicks

✗ AVOID

A developer has to navigate to the Monitoring tab, create a new monitor, set up the APL query in the 'Query' box, define the notification rule in the 'Alerting' section, and save everything manually.

✓ INSTEAD

Just ask your agent directly: 'Create a monitor for latency over 500ms on dataset production-logs.' The MCP handles all those steps automatically using `create_monitor`.

Mixing data sources without structure

✗ AVOID

A team dumps logs from three different services into one giant CSV file, making it impossible to filter or query specific metrics accurately.

✓ INSTEAD

Use the agent to ingest and define structured datasets. You can run `ingest_data` with your raw files, then use `create_dataset` to organize them logically before querying.

Forgetting to update old alerts

✗ AVOID

A service is deprecated, but the team forgets to turn off the associated monitor and notification rule, leading to constant, useless alerts.

✓ INSTEAD

Use `delete_monitor` and `delete_notifier` when services change. This keeps your monitoring stack accurate and prevents alert fatigue.

The Right Fit

Use this MCP if you need a single interface to manage the entire observability lifecycle—from data ingestion and schema definition, through complex query execution (APL), to setting up real-time monitoring alerts. You should connect it if your current workflow involves manually switching between multiple tools just to get a complete picture of system health or debug an issue.

Don't use this MCP if you only need simple data storage without querying capabilities, in which case a basic file sync tool might suffice. Also, don't use it if your primary goal is purely visual dashboarding; while `create_dashboard` helps, the true power lies in using the agent to run queries first and then visualizing the results.

If you are only interested in generating reports from static data dumps without real-time monitoring needs, a dedicated BI tool might be better. But if your job involves reacting to live system changes or debugging production issues, this MCP is necessary.

Axiom: Solving Observability Log Management with AI Agents

Right now, figuring out why a service broke usually means copy-pasting logs from one console into another. You're switching tabs to check the dataset status, running queries in a separate CLI window, and then jumping back to the dashboard just to mark where the failure started.

With this MCP, you simply ask your agent what you need—for example, 'Find all errors related to user ID 456.' The system runs the query against live data, pulls up relevant logs, and shows you the immediate answer without any manual context switching.

Axiom: Streamlining Infrastructure Monitoring with AI Agents

Manually keeping monitoring stacks current is a headache. You have to remember which dashboards need updating when a metric changes, and you must manually set up every single notification rule for every potential failure point.

Now, the agent handles it. Need an alert? Ask your agent; it executes `create_monitor` and sets up the whole pipeline instantly. It's all about speaking to the system instead of clicking through its menus.

31 Tools for Querying Log Data and Monitoring Metrics

Use these tools to list, create, update, and retrieve every component needed to monitor and analyze your production logs and metrics.

#	TOOL	DESCRIPTION
01	`create_annotation`	Adds a specific note or marker to an existing dashboard for context.
02	`create_dashboard`	Builds a new visual dashboard to track multiple system metrics simultaneously.
03	`create_dataset`	Establishes a new container in Axiom to hold and manage specific types of raw data.
04	`create_monitor`	Sets up automated checks that constantly watch for performance dips or error conditions.
05	`create_notifier`	Creates an alert system that sends notifications when a monitored metric crosses a defined threshold.
06	`delete_annotation`	Removes annotations from dashboards once the temporary context is no longer needed.
07	`delete_dashboard`	Deletes entire dashboards to keep your monitoring view clean and focused.
08	`delete_dataset`	Permanently removes a dataset, freeing up storage space for telemetry data.
09	`delete_monitor`	Turns off an automated monitor check when the service being tracked is decommissioned.
10	`delete_notifier`	Removes an alert notification rule that was previously set up for a specific event type.
11	`get_annotation`	Retrieves details about a single annotation using its unique ID number.
12	`get_dashboard`	Fetches the configuration and metrics of an existing dashboard by its UID.
13	`get_dataset`	Retrieves metadata and status for a specific dataset, confirming its existence and contents.
14	`get_monitor`	Gets the current status and settings of a specific monitoring check.
15	`get_notifier`	Retrieves details about an alert notifier rule based on its ID.
16	`get_org`	Fetches organizational information, confirming tenancy boundaries for data access.

#	TOOL	DESCRIPTION
17	`get_user`	Looks up a user's profile and associated details using their unique identifier.
18	`ingest_data`	Loads raw JSON, NDJSON, or CSV content directly into an active Axiom dataset for analysis.
19	`list_annotations`	Provides a list of all annotations currently defined across your monitored systems.
20	`list_dashboards`	Shows a catalog of every dashboard you have created or are subscribed to view.
21	`list_datasets`	Retrieves a complete list of all available datasets for log and metric storage.
22	`list_monitors`	Lists all configured monitors, showing their status and the metrics they track.
23	`list_notifiers`	Shows every active notification rule and its associated alert triggers.
24	`list_tokens`	Displays a list of API tokens currently generated for security auditing purposes.
25	`list_users`	Returns an inventory of all user accounts tied to the organization's tenancy.
26	`run_query`	Executes complex, customized APL queries against your entire Axiom data corpus.
27	`update_annotation`	Modifies the content or visibility settings of an existing annotation on a dashboard.
28	`update_dashboard`	Applies changes to metrics, visualizations, or widgets within an established dashboard.
29	`update_dataset`	Alters the schema or metadata associated with an already ingested dataset.
30	`update_monitor`	Adjusts the parameters of a monitoring check, like changing its threshold value or frequency.
31	`update_notifier`	Modifies the recipients or conditions for an existing alert notification rule.

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 available datasets for production logs.

Datasets Found:

- `production-logs` (ID: ds_123) - Last updated 5 minutes ago. Contains raw JSON records.
- `staging-metrics` (ID: ds_456) - Last updated 1 hour ago. Optimized for CSV queries.
- `security-audit` (ID: ds_789) - Active monitoring dataset, requires elevated permissions.

U Count the number of authentication failures in the last four hours.

Running APL query on `production-logs`...

Query executed: `['production-logs'] | where level == 'error' and message contains 'auth failure' | summarize count()`

Result: 78 authentication failures found in the last four hours. This is a 20% increase over the previous period.

U Create a new dashboard to track database connection pool utilization.

Dashboard 'DB Pool Health' Created.

- **Status:** Active
- **Metrics:** Connection Count (Gauge), Utilization Percentage (Graph)
- **Notes:** I've set the threshold warning at 85% and scheduled it to update every minute. Dashboard UID: dash_xyz123.

Frequently Asked Questions

01 **How does Axiom MCP help me analyze my logs without writing complex code?**

Axiom MCP lets you query your logs conversationally. Instead of remembering APL syntax, you just ask the agent what you need to know—like 'show errors from last night'—and it runs the correct analysis for you.

02 **Can Axiom MCP manage my system alerts?**

Yes. You can use this MCP to create new monitors and notifiers instantly. You just tell your agent what threshold to watch (e.g., 'if CPU > 90%'), and it handles the setup so you get real-time alerts.

03 **What kind of data can I load into Axiom using this MCP?**

You can ingest various raw formats, including JSON, NDJSON, or CSV. This means you don't have to preprocess your logs; the agent loads and prepares them for querying right away.

04 **Does Axiom MCP help with user access control?**

It does. You can easily get information about users, list all API tokens, or view organization details using this MCP. This makes auditing security compliance much faster than manual checks.

05 **Is Axiom MCP suitable for data analysts working with large datasets?**

Absolutely. It provides powerful tools to ingest massive amounts of telemetry and gives you the ability to run complex processing language queries, turning raw logs into actionable metrics.

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 `"axiom": { "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

Axiom 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 Axiom. 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	Axiom MCP
Server ID	`019e3868-39c1-7114-8520-ef07fed80569`
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/axiom.