MCP SERVER | NO CODE | CLOUD HOSTED

Azure Log Analytics Workspace MCP for AI Agents

Analyze System Performance and Health with Scoped KQL Queries

Azure Log Analytics Workspace MCP provides secure, scoped access to a single Azure Log Analytics table. It lets your AI client execute complex KQL queries directly against critical system logs. This is perfect for debugging applications or analyzing performance spikes without needing global permissions.

F Quality Score 3.6/100

kql | log-querying | cloud-monitoring | telemetry | troubleshooting | scoped-access

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/azure-log-analytics-workspace

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.

Azure Log Analytics Workspace MCP

1 tools available
Cloud-hosted on Vinkius

Debugging production issues often means digging through massive amounts of log data. Normally, this requires jumping between dashboards and running multiple manual searches—a process that's slow and prone to missing key details. This MCP changes that by giving your AI agent one surgical capability: the ability to run Kusto Query Language (KQL) queries on a single, designated Log Analytics table. Critically, it doesn't grant global access; its scope is tightly contained. This safety feature means you can safely troubleshoot application errors or analyze traffic patterns without risking exposure to sensitive audit trails across your entire Azure environment. You simply provide the necessary KQL operations—for example, filtering by a time range or specific error codes—and your agent handles the rest. It's a secure way to get deep observability right where you need it.

Core Capabilities

01 — Execute Kusto Query Language (KQL) queries

The AI client runs complex, filtered searches against the designated Log Analytics table.

02 — Filter log data by time or severity

You can narrow down results to specific time windows or only show records flagged with errors.

03 — Extract structured insights from raw logs

The agent parses complex JSON payloads within the logs to pull out specific data points, like user IDs or request statuses.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/azure-log-analytics-workspace** — connect your AI agent in three steps.

01 You ask your AI client a question about system performance (e.g., 'Show me all 500 errors from the last hour').

02 Your agent translates that request into specific KQL operations and sends them to this MCP.

03 The MCP executes the query against the single authorized log table and returns the filtered, structured results to your AI client for interpretation.

The bottom line is, you talk naturally about the data you need, and the system handles the complex querying process.

Built For

This MCP is essential for any operational team dealing with live cloud infrastructure. Think SREs who get frustrated manually clicking through Azure dashboards at 2 a.m., or security analysts needing quick, scoped access to investigate incidents without excessive permissions.

Site Reliability Engineer (SRE)

Running deep-dive queries to isolate the root cause of an intermittent production failure, saving hours of manual dashboard pivoting.

DevOps Engineer

Monitoring deployments after a code push by checking for specific error patterns or unexpected resource utilization spikes in real time.

Security Analyst

Investigating potential breaches by querying logs for unusual user access attempts or activity that falls outside normal operational parameters.

What Changes When You Connect

01 Pinpoint the exact moment an issue started. Instead of sifting through terabytes of data, you run a precise query to find only relevant error logs.

02 Eliminate permission creep risks. Because this MCP is locked down to a single table, your agent can debug without ever touching sensitive global audit records.

03 Speed up incident response. Your AI client executes complex KQL syntax—like joining time filters with severity levels—in seconds, giving you instant context.

04 Go beyond simple text searches. The tool supports parsing JSON payloads, letting the AI extract metrics like specific request IDs or user session details.

05 Use structured query language (KQL) directly through natural conversation. No more learning complicated command-line syntax just to check logs.

Real-World Applications

Debugging a User Authentication Failure

A user reports they couldn't log in this morning. Instead of checking ten different services, your agent runs a query targeting failed login attempts over the last 4 hours and pulls out the specific error code and associated user ID.

Investigating Traffic Spikes

The application suddenly slowed down yesterday afternoon. Your agent queries logs to compare traffic volume (requests per second) during the slow period versus a normal baseline, pinpointing the exact time of degradation.

Auditing Specific Resource Activity

You need to know who accessed a specific database resource on Monday. Your agent queries logs for entries containing that unique resource ID and filters by user role, giving you a clean list of access attempts.

Patterns to Avoid

Asking the AI for 'all' logs

✗ AVOID

Prompting your agent with simply, 'Show me the logs.' This will either fail due to scope limits or return a massive, unusable dump of data.

✓ INSTEAD

Always tell your agent exactly what you need. Use time constraints and filters: 'Query all records where severity level is Error AND TimeGenerated is greater than 1 hour ago.'

Trying to join multiple tables

✗ AVOID

Thinking the MCP can search across logs from both the networking table and the application table simultaneously.

✓ INSTEAD

This MCP only accesses one single, scoped log table. You must filter all your necessary data points within that specific table using KQL.

Forgetting to specify a time range

✗ AVOID

Running a query without `| where TimeGenerated > ago(24h)` and getting results from the last year.

✓ INSTEAD

Always scope your queries with explicit time filters. Specify 'last 3 hours' or 'yesterday only' right in your prompt to keep results manageable.

The Right Fit

Use this MCP if you have a specific, known log table and need precise, secure access to its data without the risk of global permissions. You should use it when troubleshooting an incident or analyzing structured telemetry where filtering by time or severity is key. Don't use it if you need to pull data from multiple disparate services; that requires connecting several different MCPs. Also, don't use it if your primary goal is searching unstructured documents; this tool only handles structured log records.

Azure Log Analytics Workspace MCP: Solving Production Monitoring Pain Points with KQL

Today, investigating a production issue means logging into the Azure portal, finding the right Log Analytics workspace, and then manually running dozens of queries. You're copying timestamps from one dashboard, pasting them into another query to narrow down the search, and constantly refreshing pages just to piece together what went wrong.

With this MCP, you talk to your agent like talking to a teammate. Instead of clicking through tabs or manually adjusting date ranges, you simply ask: 'What were the top 10 error codes from yesterday between 2 PM and 3 PM?' Your agent executes that complex KQL operation instantly, giving you a clean, actionable list of results.

Azure Log Analytics Workspace MCP: Analyzing Incident Response with Scoped Querying

Manual incident response is slow. It involves checking for correlations across different log sources—network logs, application logs, identity logs—and trying to figure out which data points belong together just by looking at timestamps.

This MCP centralizes the querying process on a single table while retaining full KQL power. You get immediate, surgical insight into event sequences and performance bottlenecks without ever needing global access keys or juggling multiple interfaces.

Azure Log Analytics Workspace: 1 Tool for Cloud Monitoring and KQL Querying

Use the available tool to execute powerful Kusto Query Language queries against a single Azure log table, retrieving specific operational insights.

#	TOOL	DESCRIPTION
01	`query_logs`	Runs a KQL query against the configured Log Analytics table using only the operations you specify.

See It in Action

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

U What were the top 5 error codes that occurred in the last two hours?

Top 5 Error Codes (Last 2 Hours)

CODE	COUNT
AUTH_FAIL_401	45
DB_TIMEOUT_503	18
RESOURCE_LIMIT	9
INPUT_VALIDATION	7
NETWORK_DROP	6

*Query used: `| where TimeGenerated > ago(2h) and SeverityLevel == 'Error' | summarize count() by Code | top 5 by count_`

U Show me all requests from a specific user ID that failed with a timeout error.

Timeout Errors for User U-901

- Total failures found: 3
- Time Window: 2024-06-15 10:00 to 10:05 UTC

TIMESTAMP	REQUEST ID	STATUS CODE
10:01:15	req-a7d3	503
10:02:40	req-b9e1	503
10:04:55	req-c2f8	503

U Find out how many times the 'admin' user was active in the last day.

User Activity Summary (Admin)

- **Total Records:** 1,420 logs.
- **Key Metrics:**

* Successful logins: 1,395
* Failed attempts: 25
* Most common action: Data retrieval via API call.

This suggests the user is highly active but had a few minor authentication hiccups.

Frequently Asked Questions

01 **How do I use the Azure Log Analytics Workspace MCP for debugging?**

You simply ask your agent what you're looking for—for instance, 'Show me all network connection failures from yesterday.' The MCP handles the complex KQL query and returns a clean table of results.

02 **Does this MCP work with different types of logs?**

It works on structured log data within one specific Azure Log Analytics table. You need to know roughly what kind of data is in that table (e.g., application events, security records) to ask the right question.

03 **Is this safe for my production environment?**

Yes, safety was the main design focus. The MCP only allows querying a single, specified log table, which means your agent can't accidentally access sensitive logs elsewhere in Azure.

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 `"azure-log-analytics-workspace": { "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

Azure Log Analytics Workspace 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 Azure Log Analytics Workspace. 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	Azure Log Analytics Workspace MCP
Server ID	`019e386a-1aed-70df-afca-8074060a9f66`
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/azure-log-analytics-workspace.