MCP SERVER | NO CODE | CLOUD HOSTED

Databricks MCP for AI Agents

Monitor Data Lakehouse Cluster Health & Job Status

Databricks MCP connects your agent directly into your data intelligence platform. You can audit SQL warehouses, list compute clusters, track complex job executions, and explore structured data across Unity Catalog without leaving your chat window. It gives full control over your lakehouse orchestration via conversation.

A+ Quality Score 100/100

lakehouse | data-engineering | cluster-management | sql-warehousing | data-governance | big-data

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

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.

Databricks MCP

8 tools available
Cloud-hosted on Vinkius

You're managing a massive data lakehouse, but checking status means jumping between dashboards, running manual queries, and copying logs. This MCP lets you talk to your platform instead. You can ask your agent to list all active compute clusters or check the recent run history for a specific ETL job just by asking. Need to know where your structured data lives? Your agent will query the Unity Catalog and map out every root catalog and schema. It’s about getting instant, auditable visibility into everything running on your platform. Because Vinkius hosts this MCP, you connect once from any compatible client—Claude, Cursor, or Windsurf—and get immediate access to complete data governance oversight.

Core Capabilities

01 — Audit and manage compute clusters

List all active nodes and retrieve deep details on specific clusters' current health and capacity limits.

02 — Track job pipelines and workflows

See every configured workflow, list jobs, and monitor recent executions to verify data pipeline status or find failure points.

03 — Govern structured data locations

Identify where your data lives by listing root Unity Catalog catalogs and detailed schemas across the workspace.

04 — Manage SQL data warehousing resources

Enumerate all configured SQL Serverless warehouses and track their current operational boundaries for cost control.

05 — Verify user permissions and identity

Fetch profile information for the authenticated user or service principal to audit active workspace permissions.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius.

02 Input your Databricks Host URL and Personal Access Token (PAT) into your agent client.

03 Start asking questions. Your agent uses the connected tools to perform audits, list resources, or check job statuses in natural language.

The bottom line is that you manage your data lakehouse by talking to it, making complex operations simple prompts.

Built For

This MCP is built for the technical teams who live in the data platform. If you're tired of switching between dashboards and manual API calls just to check if a job failed or what resources are running, this is for you.

Data Engineer

You use it to monitor job runs and cluster health without ever leaving your development environment.

Analytics Engineer

You check Unity Catalog schemas and verify SQL warehouse availability in real-time for new data models.

MLOps Engineer

You track model training jobs and audit compute cluster configurations to ensure proper resource allocation.

What Changes When You Connect

01 Audit cluster health instantly. You can use `get_cluster` to get detailed specifications or `list_clusters` to see the full inventory of nodes running in your workspace.

02 Never miss a failed pipeline run again. By listing job runs and using `list_job_runs`, you pinpoint exactly where data workflows break, saving hours of manual debugging.

03 Manage costs by visibility. You can list all SQL warehouses (`list_warehouses`) to track which serverless resources are active and consuming credits right now.

04 Understand your data map. Instead of guessing where a table is, use `list_catalogs` and `list_schemas` to get an auditable inventory of every piece of structured data.

05 Control access rights. You can run the identity check (`get_me`) to verify if the service principal currently running the job has the necessary permissions.

Real-World Applications

Debugging a failed ETL pipeline

A data engineer asks their agent, 'What went wrong with the Daily Sales ETL?' The agent calls `list_job_runs`, identifies the failing job run ID, and reports that the error was due to an upstream cluster timeout. They then use `get_cluster` to check if resource limits were hit.

Auditing data governance for compliance

An analytics engineer needs proof of all structured data sources. The agent uses `list_catalogs` and then iterates through `list_schemas`, providing a complete, auditable map of the entire Unity Catalog structure.

Resource optimization before scaling

An MLOps engineer wants to know if they can afford more compute power. They run `list_clusters` and compare the active count against the usage reported by `get_cluster`, determining exactly which clusters need adjustment.

Verifying data access permissions

A platform team member needs to know if a new service account has full visibility. They run `get_me` and audit the returned profile information against required workspace roles, confirming proper identity oversight.

Patterns to Avoid

Checking status via dashboards

✗ AVOID

Opening the web UI, navigating to 'Jobs,' then running reports for job runs. This takes minutes and requires multiple clicks.

✓ INSTEAD

Ask your agent directly: 'Show me the last five runs for the Daily ETL job.' The agent uses `list_job_runs` and provides the summary instantly.

Guessing data location

✗ AVOID

A new hire asks, 'Where is the customer table?' They spend an hour asking teammates until someone manually points them to the right catalog.

✓ INSTEAD

Ask your agent: 'List all root catalogs in Unity Catalog.' The agent uses `list_catalogs` and gives you the full list instantly.

Ignoring resource boundaries

✗ AVOID

Launching a job without checking if an existing SQL warehouse is already active, leading to unexpected cost spikes.

✓ INSTEAD

Ask your agent: 'What are my currently configured SQL warehouses?' The agent uses `list_warehouses` and confirms the operational status.

The Right Fit

Use this MCP if you need full visibility into the technical operations of a data lakehouse. Specifically, connect it when auditing job runs (using `list_job_runs`), monitoring compute resource usage (`get_cluster`), or mapping out your structured data landscape (`list_catalogs`). Don't use this if your only need is to view marketing reports or analyze business metrics; those require a BI tool connection. If you just want a simple list of users, that might be better handled by an identity management MCP instead.

Databricks MCP for AI Agents: Auditing Data Lakehouse Job Runs

Right now, checking the health of your data pipelines is a manual nightmare. You have to navigate through job orchestration dashboards, find the specific job ID, and then scroll through logs until you locate the failure point or confirmation that everything ran successfully. This process involves opening multiple tabs and copying error codes just to report the status.

With this MCP, you simply ask your agent, 'What happened with the nightly inventory pipeline?' The agent calls `list_job_runs` and provides a clean summary: Job ID 987 succeeded at 6 AM. Run 985 failed due to X error. You get immediate answers and actionable data points without touching a dashboard.

Databricks MCP for AI Agents: Governing Unity Catalog Schemas

If you don't know exactly where your structured data lives, governance is impossible. Today, finding all related datasets requires running several manual queries across different catalog views and cross-referencing team documentation to map the schema locations.

Now, just ask: 'Show me every database in the main catalog.' The agent uses `list_schemas` and instantly outputs a structured list of every available dataset. You gain immediate, definitive knowledge of your entire data inventory.

8 Tools for Databricks Data Lakehouse Management

Use these tools to list everything from active clusters and jobs to the entire catalog structure within your data lakehouse environment.

#	TOOL	DESCRIPTION
01	`list_clusters`	Retrieves a full list of all compute clusters configured in your Databricks workspace.
02	`get_cluster`	Fetches detailed operational information for a specific cluster ID or name.
03	`list_jobs`	Lists every configured data workflow and job that runs across your platform.
04	`list_job_runs`	Provides a history of all executed jobs, showing success or failure status for specific workflows.
05	`list_warehouses`	Enumerates every SQL Serverless warehouse configured in your environment.
06	`list_catalogs`	Lists all root catalogs defined within the Unity Catalog structure.
07	`list_schemas`	Retrieves a list of databases or schemas contained within a specified catalog.
08	`get_me`	Identifies the current user's profile and active permissions in the Databricks workspace.

See It in Action

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

U List all compute clusters for my team so I can check their status.

Here are the 4 active clusters in the workspace:

- **Standard-ETL-Cluster:** Running (Nodes: 6). Last updated: 5 minutes ago.
- **ML-Training-Node:** Terminated. Reason: Job completion.
- **Shared-Dev-Pool:** Running (Nodes: 3). High usage detected.
- **Finance-Audit-Cluster:** Stopped. Needs manual restart.

U I need to know which SQL warehouses are currently running and how many there are.

I found three active, serverless data warehouses:

WAREHOUSE NAME	STATUS	LAST USED	PURPOSE
Prod_Reporting	Active	10 mins ago	BI Dashboards
Dev_Sandbox	Active	Now	Testing Queries
Historical_Arch	Idle	Never	Read-only access

You can monitor the boundaries of these resources directly.

U What were the results for my last three data jobs?

Checking job history... Here's a quick overview:

- **Job: Daily Sales ETL (ID 987):** Success. Finished at 6:00 AM.
- **Job: User Sync (ID 985):** Failure. Timeout error detected. Check logs for details.
- **Job: Inventory Update (ID 984):** Success. Completed quickly, ran in 12 minutes.

Frequently Asked Questions

01 **How does the Databricks MCP help me track my cluster usage?**

The Databricks MCP lets you list all compute clusters and get detailed information on specific nodes. This means you can audit which resources are running, check their health, and understand your overall capacity limits without logging into the platform.

02 **Can I use this MCP to see if my data pipelines ran correctly?**

Yes. You can list all configured jobs and monitor job runs. Your agent checks the status of past executions, telling you immediately which workflows succeeded or failed, and why.

03 **Does the Databricks MCP help with data governance in Unity Catalog?**

Absolutely. You can list root catalogs and then drill down to find all schemas within them. This gives you a full inventory map of where every piece of structured data resides, which is key for compliance.

04 **What if I need to verify my user permissions in Databricks?**

The MCP has an identity oversight tool that fetches your profile information. This lets you confirm exactly what roles and permissions are active for the service principal running your workflow, which is critical for security audits.

05 **Is this better than checking status on a dashboard?**

Yes. Instead of manually clicking through multiple dashboards, you ask your agent a question, and it executes the necessary checks (like listing job runs or warehouses) and gives you a summarized answer instantly.

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 `"databricks": { "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

Databricks 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 Databricks. 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	Databricks MCP
Server ID	`019d7581-72d8-72a2-88bb-98232613173b`
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/databricks.