MCP SERVER | NO CODE | CLOUD HOSTED

Tinybird Data Platform MCP

Audit, query, and manage real-time data flows in chat.

Tinybird Data Platform connects your AI agent directly to a real-time data warehouse, giving you hands-on control over complex analytics infrastructure. You can manage all your data sources and transformation logic from chat alone. Use this MCP to list every available workspace, check row counts for any data source, audit pipeline status, or run arbitrary SQL queries against live data. It's full operational oversight for data engineers.

A+ Quality Score 100/100

real-time-data | data-ingestion | sql-queries | data-pipelines | api-endpoints | analytics-infrastructure

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/tinybird-data-platform

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.

Tinybird Data Platform MCP

10 tools available
Cloud-hosted on Vinkius

This MCP lets you treat your entire analytical infrastructure like a conversational service. Instead of opening the Tinybird dashboard and clicking through five different tabs just to check if a data stream is healthy, your agent handles it all. You can ask it to list every workspace available or inspect a specific Data Source's current ingestion stats—all in natural language. It lets you run complex SQL queries without writing boilerplate connection code yourself. Furthermore, the platform allows deep dives into how your data changes, enabling you to retrieve the exact SQL logic used by any Pipe and even execute those published transformations for instant results. Connecting this MCP through Vinkius means you get centralized control over all these critical functions from one place, making operational monitoring feel like a simple chat conversation.

Core Capabilities

01 — Inventory Data Sources

List every data source and workspace available in your current analytical environment.

02 — Audit Data Source Status

Get detailed information, including row counts and storage sizes, for any specified data source.

03 — Examine Data Pipelines

List all transformation pipelines (Pipes) or retrieve the specific SQL logic used within them.

04 — Run Ad-Hoc Queries

Execute any arbitrary SQL query against your live data warehouse using ClickHouse dialect.

05 — Analyze Pipeline Execution

Run a specific Pipe and immediately retrieve the results as structured JSON output.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/tinybird-data-platform** — connect your AI agent in three steps.

01 Subscribe to this MCP on Vinkius and provide your Tinybird Admin Token.

02 Ask your agent to perform an action, such as listing all available data sources or checking the stats for a specific Data Source.

03 The agent executes the necessary command, pulls the real-time metrics, and presents the results directly in chat.

The bottom line is you manage complex data flows using simple conversation prompts.

Built For

Data Engineers and Analytics Leads need this. They're tired of context-switching between the dashboard, SQL IDEs, and monitoring tools just to validate a single query or audit a failing pipeline. This MCP puts full operational visibility into your chat window.

Data Engineer

Audits pipe logic using list_pipe_nodes to understand transformation dependencies, and runs execute_sql_query for quick validation during development.

Product Analyst

Checks data source stats using get_datasource_stats or executes a query_pipe_data command without having to navigate the entire analytics platform.

DevOps Team Lead

Monitors ingestion performance and token scopes by calling list_auth_tokens, ensuring all pipelines are running within scope.

What Changes When You Connect

01 Stop guessing if your data is fresh. Use get_datasource_stats to immediately check row counts and ingestion usage for any source without opening a dashboard tab.

02 Debugging complex pipelines just got easier. You can use list_pipe_nodes or get_pipe_details to see the exact SQL logic used in any transformation, accelerating development time.

03 Run queries instantly. Use execute_sql_query to fire off ad-hoc analytical requests and explore your data directly via natural language command.

04 Full infrastructure visibility. Need to know what workspaces exist? list_workspaces gives you a quick inventory of everything connected.

05 Test transformations safely. Instead of running an entire Pipe in the GUI, use query_pipe_data to execute it and get the structured JSON output immediately.

Real-World Applications

Checking data health on a Friday afternoon

A product analyst needs to know if the user event stream has been updated since last night's deployment. Instead of clicking into the 'user_events' source and scrolling through metrics, they just ask their agent to get_datasource_stats. The agent instantly replies with the latest row count and ingestion status.

Debugging a broken data pipeline

A data engineer finds that the 'monthly_report' pipe is failing intermittently. They use list_pipe_nodes to see if the SQL logic changed, then run get_datasource_details on the upstream source to verify field types, pinpointing the failure point immediately.

Quickly proving a hypothesis

A manager wants to know how many users came from 'partner X' last month. Instead of building a new dashboard and waiting for approval, they run an execute_sql_query command directly through the agent, getting the answer in seconds.

Auditing security access

The DevOps team needs to verify which services have access to sensitive data. They use list_auth_tokens to generate a full audit trail of all active tokens and check their scopes, ensuring least privilege is maintained.

Patterns to Avoid

Treating the MCP like an API wrapper

✗ AVOID

Sending repetitive commands like 'List Data Sources. Now get stats for Data Source A. Then list Pipes.' This requires multiple steps and context switching.

✓ INSTEAD

Ask your agent to consolidate: 'Give me a full audit of data sources, including their current row counts and listing all associated pipes in the workspace.' The MCP handles the sequence.

Over-engineering simple queries

✗ AVOID

Wasting time creating complex views or new dedicated dashboards just for one simple metric.

✓ INSTEAD

Use execute_sql_query. If you need a quick number, run it directly with the agent instead of building permanent infrastructure.

Assuming data completeness

✗ AVOID

Running query_pipe_data and assuming the result set is perfect without checking the source.

✓ INSTEAD

Always preface pipe execution by calling get_datasource_stats first. This verifies that the underlying Data Source has sufficient, up-to-date metrics.

The Right Fit

Use this MCP if your core problem is accessing and managing real-time data infrastructure through chat conversation. If you need to run ad-hoc queries (execute_sql_query) or audit the state of multiple connected services (list_datasources, list_auth_tokens), this is built for you. Don't use it if your only goal is basic file storage management; that requires a different type of integration. Similarly, don't rely on it to fix bad data quality—it only reports what the data *is*. If you just need to build an entirely new reporting layer from scratch without querying existing sources, look for ETL tools rather than this MCP.

The Pain of Context-Switching Data Checks

Today, checking the health of your analytics platform is a multi-step chore. You open the main dashboard to check Data Source A's row count. Then you tab over to the Pipelines section to see if Pipe B has run recently, and finally, you might have to copy a token ID into a separate terminal window just to audit the permissions. It's clicking, switching tabs, and constantly copying credentials.

With this MCP, all that operational visibility lives inside your chat window. You simply ask your agent to get_datasource_stats for Source A while simultaneously listing_pipes to check Pipe B's status. The agent gathers all that disparate information and presents it back in one clean, conversational response.

Tinybird Data Platform MCP: Querying Live Insights

Gone are the days of manually running a query in a separate client just to validate if the data is ready. You can now use execute_sql_query and then immediately follow up with query_pipe_data on the resulting dataset, all within the same chat session.

The result is a continuous flow of information, making your entire analytical workflow feel less like software operations and more like having an expert teammate sitting right next to you.

Tinybird Data Platform with 10 Tools

Use these tools to manage everything from listing workspaces to executing complex real-time SQL queries against your entire data platform.

#	TOOL	DESCRIPTION
01	`execute_sql_query`	Runs an arbitrary SQL query directly against the Tinybird workspace data.
02	`get_datasource_details`	Retrieves detailed information about a specific Data Source in the platform.
03	`get_pipe_details`	Gets comprehensive details for any specified data transformation Pipe.
04	`get_datasource_stats`	Pulls current ingestion and usage statistics for a selected Data Source.
05	`list_datasources`	Retrieves a full list of every data source connected to the workspace.
06	`list_pipe_nodes`	Lists all individual SQL nodes contained within a specific transformation Pipe.
07	`list_pipes`	Retrieves a list of every available data transformation pipe in the workspace.
08	`list_auth_tokens`	Lists all authentication tokens, allowing you to audit access control within the workspace.
09	`list_workspaces`	Retrieves a list of every available data workspace across your account.
10	`query_pipe_data`	Executes an entire Pipe and returns the resulting dataset as structured JSON.

See It in Action

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

U List all data sources in my Tinybird workspace.

Fetching data sources... I found 3 sources: 'events_log' (1.2M rows), 'users_data' (50k rows), and 'api_analytics' (800k rows). Which one would you like to inspect?

U Run the pipe 'monthly_revenue_summary' with limit 5.

Executing pipe... Here are the top 5 records from 'monthly_revenue_summary': 1. Jan ($12k), 2. Feb ($15k), 3. Mar ($11k)...

Frequently Asked Questions

01 **How does Tinybird Data Platform MCP list all data sources?**

You use the list_datasources tool. It immediately provides a comprehensive inventory of every source connected to your workspace, giving you an at-a-glance view of what's available for analysis.

02 **Can I check my data pipeline status using Tinybird Data Platform MCP?**

Yes. You can use list_pipes to see every defined pipe, and then get_pipe_details or query_pipe_data to understand its logic or execute it for results.

03 **What is the difference between execute_sql_query and query_pipe_data?**

execute_sql_query runs any SQL you write, giving maximum flexibility. query_pipe_data executes a pre-built Pipe, ensuring that your logic follows established data transformation rules.

04 **Does Tinybird Data Platform MCP help with security audits?**

Absolutely. You can use list_auth_tokens to retrieve a full list of all authentication tokens and audit who has access across the workspace.

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 `"tinybird-data-platform": { "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

Tinybird Data Platform 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 Tinybird Data Platform. 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	Tinybird Data Platform MCP
Server ID	`019d848e-f293-7286-96e3-4cf81e16a375`
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/tinybird-data-platform.