MCP SERVER | NO CODE | CLOUD HOSTED

Hevo Data MCP

Monitor ETL Flow & Usage Via Chat Commands

Hevo Data (ETL & Data Pipeline) lets you manage your entire data integration stack using natural conversation. List pipelines, check destination status across BigQuery or Snowflake, and audit row usage without jumping between dashboards. Take full control of your automated ETL orchestration directly from your AI client.

A+ Quality Score 100/100

etl | data-pipelines | data-warehousing | data-integration | pipeline-monitoring | automated-sync

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/hevo-data-etl-data-pipeline

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Hevo Data (ETL & Data Pipeline) MCP

6 tools available
Cloud-hosted on Vinkius

Managing complex data flows usually means opening five different tabs: one for pipeline status, another for billing metrics, a third to check if the data hit BigQuery, and so on. This MCP changes that by giving you direct conversational access to your Hevo Data account. You can ask your AI client simple questions—like 'Are my Snowflake destinations healthy?' or 'How many rows did I use this month?'—and get immediate answers. It lets you orchestrate pipelines and monitor every connection, from the transformation models defining your logic to the final billing usage report. If you're building your agent catalog on Vinkius, adding this MCP means your users can manage mission-critical data assets without ever leaving their chat window. You simply tell your AI client what you need, and it executes the checks across all your connected data destinations.

Core Capabilities

01 — List active pipelines

Retrieves a list of every automated ETL pipeline configured in your account.

02 — Check destination health

Analyzes the status and connection details for all data warehouses like BigQuery, Snowflake, or Redshift.

03 — Audit account usage

Pulls real-time metrics on your row replications and overall billing usage against your quota.

04 — List transformation models

Shows the specific mappings and logic attached to keep your data quality consistent.

05 — Discover workflow connections

Maps out complex, multi-step data workflows connecting different transformations across your stack.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/hevo-data-etl-data-pipeline** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your unique Hevo Data API key and region (e.g., US or EU).

02 Connect the credentials to your preferred AI client, like Cursor or Claude.

03 Ask a natural language question—for example, 'Check my data pipeline usage'—and get immediate, actionable reports.

The bottom line is you manage complex, critical data infrastructure using only chat commands.

Built For

Data Engineers who are tired of clicking through dashboard tabs to find a single replication failure. Analytics Leads who need to validate transformation logic before reporting starts. Operations Teams needing instant billing checks without logging into the console.

Data Engineer

Uses this MCP to monitor ETL pipeline health and destination replication statuses instantly, preventing manual dashboard hopping.

Analytics Lead

Checks transformation models and workflow orchestrations via chat to guarantee data is clean and ready for reporting.

Operations Manager

Tracks row usage, account billing ceilings, and overall pipeline health in real time to keep the organization within budget.

What Changes When You Connect

01 Stop jumping between dashboards. You can list pipelines and check destination status—all in one conversation.

02 Get an instant audit of your account usage by calling get_usage, so you never exceed your row quota unexpectedly.

03 Verify data integrity using list_models to track the exact mappings bounding your staging logic for quality assurance.

04 Map out complex connections using list_workflows. You'll see how multiple transformations link across your entire data stack.

05 Quickly understand which sources feed into which targets by listing all destinations, ensuring no critical warehouse is missed.

Real-World Applications

The nightly sync failed; I need to know why.

An Ops Manager asks the agent: 'What's wrong with my data flow?' The agent calls get_pipeline and list_destinations, reporting that the pipeline is down because the Snowflake destination connection timed out. The manager fixes it immediately without logging into any web UI.

I need to audit our billing before Q3.

An Analytics Lead queries: 'How many rows did I use this month?' The agent calls get_usage, providing a usage breakdown and projection. This prevents unexpected overages when the data team scales up reporting.

We added a new staging area; where does it go?

A Data Engineer uses list_destinations to confirm that their new Redshift cluster is correctly recognized by Hevo. They then use get_pipeline to ensure the required sync job is configured for that specific target.

I need to prove data lineage for an audit.

The agent uses list_workflows combined with list_models, generating a map of every transformation step and mapping. This proves exactly how raw source data becomes final report metrics for compliance checks.

Patterns to Avoid

Checking status via multiple tabs

✗ AVOID

Logging into the Hevo dashboard, clicking 'Pipelines', then opening a second tab for 'Destinations', and finally navigating to 'Usage' just to piece together one picture.

✓ INSTEAD

Ask your agent directly: 'List my pipelines and check usage.' The MCP handles calling list_pipelines and get_usage sequentially, giving you the full picture in chat.

Assuming data quality is fine

✗ AVOID

Relying only on a successful sync notification without checking if the transformation logic was correctly applied to the new source field.

✓ INSTEAD

Use list_models to verify that the correct mappings are attached, ensuring your staging logic maintains high data quality before the data reaches its destination.

Copying IDs manually

✗ AVOID

Getting a list of 50 pipelines and having to copy each unique ID one by one into another system for tracking or reporting.

✓ INSTEAD

The MCP gives you all the necessary data programmatically. You can ask it to summarize usage trends across multiple pipelines using the tools rather than manually extracting IDs.

The Right Fit

Use this MCP if your primary pain point is managing complex, multi-stage data movement and monitoring its health—you need a single pane of glass that speaks conversationally. You must be focused on ETL orchestration, tracking row usage, or validating connections across tools like BigQuery and Snowflake. Don't use this if you just need to run simple ad-hoc SQL queries against raw tables (a dedicated SQL client is better). Also, don't use it if your main goal is creating reports for end-users; that requires a BI tool. This MCP's strength lies in the operational layer: checking *if* the data moved and *how much* was used.

The Dashboard Overload

Right now, reviewing your data infrastructure means opening a dozen browser tabs. You click to see if the pipeline ran; you switch tabs to check destination status; and then you open a third tab just to verify billing limits. Copying IDs, cross-referencing dates, and figuring out which dashboard tells you the final story takes hours.

With this MCP, all that manual clicking is gone. You ask your agent what's wrong with the data flow, and it executes checks for you across multiple systems using tools like list_pipelines and list_destinations. You get one concise answer telling you exactly where the break is.

Get Full Visibility With Hevo Data (ETL & Data Pipeline)

You no longer need to manually check if a new transformation model has been attached correctly or if your row usage metrics are spiking. The MCP handles running get_usage and list_models automatically in response to your chat request.

It's instant, accurate data governance. You know the status of every part of your stack by simply asking.

Hevo Data (ETL & Data Pipeline) with 6 Tools

Use these tools to check pipeline status, track resource consumption, list connections, and monitor your overall data integration health.

#	TOOL	DESCRIPTION
01	`list_pipelines`	Lists all active data pipelines currently running.
02	`get_pipeline`	Retrieves specific details about a single pipeline.
03	`list_destinations`	Shows you every connected data warehouse destination (e.g., BigQuery, Snowflake).
04	`list_models`	Retrieves a list of all defined transformation models.
05	`list_workflows`	Lists the complex workflows that connect multiple data transformations together.
06	`get_usage`	Reports on your account's current usage metrics and billing limits.

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 active Hevo pipelines

I've found 6 active pipelines. Highlights include 'Stripe to BigQuery' (ID: pipe-001), 'HubSpot to Snowflake' (ID: pipe-005), and 'Shopify to Redshift' (ID: pipe-008). Would you like to check the usage for any of these?

U Show me the destinations for my 'Sales Data' pipeline

The 'Sales Data' pipeline (ID: pipe-005) is currently replicating to 2 destinations: a BigQuery production dataset and a Snowflake staging warehouse. Both are showing 'Healthy' status. Would you like to see the last sync time?

U How much of my row quota have I used this month?

You've used 12.5 million rows out of your 20 million row monthly quota (62.5%). At current ingestion rates, you are projected to finish the month at 18.2 million rows. I can break down usage by pipeline if needed.

Frequently Asked Questions

01 **How does Hevo Data (ETL & Data Pipeline) MCP help with billing?**

You call get_usage to instantly check how many rows you've replicated and what your remaining quota is. This prevents unexpected overages by keeping usage metrics visible in the chat.

02 **Can I list all my pipelines using Hevo Data (ETL & Data Pipeline) MCP?**

Yes, calling list_pipelines gives you a full rundown of every automated ETL pipeline configured in your account right from the agent interface.

03 **Does this MCP work with Snowflake and BigQuery?**

It monitors destinations for major data warehouses like Snowflake, BigQuery, and Redshift. You can list_destinations to confirm connectivity across all these systems.

04 **What is the difference between get_pipeline and get_usage?**

get_pipeline gives specific details on a single data flow's configuration, while get_usage reports generalized account metrics like total row replication count and billing limits.

05 **Is this useful for checking my transformation logic?**

Yes. Use list_models to review the explicit mappings that define your staging data logic and ensure quality standards are met before reporting.

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 `"hevo-data-etl-data-pipeline": { "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

Hevo Data (ETL & Data Pipeline) 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 Hevo Data (ETL & Data Pipeline). 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	Hevo Data (ETL & Data Pipeline) MCP
Server ID	`019d75b0-7b79-706a-bf46-9132f0b854df`
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/hevo-data-etl-data-pipeline.