MCP SERVER | NO CODE | CLOUD HOSTED

Coalesce MCP for AI Agents

Manage Snowflake Data Pipelines and Transformations in Chat

Coalesce MCP gives your AI agent direct control over Snowflake data pipelines. It lets you list all environments, check job status, and trigger complex transformations right from chat. You manage your entire ETL workflow without touching a UI.

F Quality Score 3.6/100

data-transformation | snowflake | etl-pipeline | data-engineering | job-monitoring | column-aware

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

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.

Coalesce MCP

8 tools available
Cloud-hosted on Vinkius

This MCP connects your AI client directly to Coalesce, the platform that manages data transformation for Snowflake. Instead of opening multiple dashboards or writing boilerplate API calls, you talk to your agent and tell it exactly what needs transforming.

Need to know if yesterday's run failed in staging? Ask your agent; it checks the job status immediately. Want to test a new pipeline on demand? Your agent triggers that specific transformation for you. You can list out every environment configured, from development through production, and inspect their current settings.

It’s all about making data governance visible via natural language. If you're working with complex Snowflake pipelines, this MCP lets your AI client manage those transformations and monitor jobs without needing the Coalesce UI open. It integrates into your existing toolset; just connect it through Vinkius and let your agent handle the heavy lifting.

Core Capabilities

01 — List all configured environments

It retrieves a complete list of every development, staging, or production environment set up in your Coalesce organization.

02 — Get specific environment details

You can pull detailed configurations for any single environment to verify settings before making changes.

03 — Check job run status and logs

Your agent checks the current progress of a pipeline run, providing real-time updates or viewing failure logs.

04 — List all available jobs

It provides an inventory of all data transformation jobs, allowing you to filter by environment or job type.

05 — Trigger a new pipeline run

You tell your agent which environment and what job to use, and it starts the required data transformation immediately. You can also specify nodes to narrow the scope of the run.

One Click on Vinkius — From Prompt to Execution

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

01 Connect this MCP to your preferred AI client via Vinkius, providing the necessary API token credentials.

02 Tell your agent what you need done using plain English. For example: 'What is the status of the Production environment run?'

03 The agent executes the correct tool call against Coalesce and presents the structured data—like job IDs or progress percentages—back to you in conversation.

The bottom line is, your AI client handles all the API communication; you just talk to it like talking to a teammate.

Built For

This MCP is for data professionals—Analytics Engineers and Data Team Leads who spend too much time clicking through dashboards. If checking job health or starting urgent pipelines is part of your routine, you need this.

Data Engineer

You use it to trigger transformations and monitor pipeline runs instantly, skipping the manual steps of opening the Coalesce UI.

Analytics Engineer

You check job statuses and debug failed data transformations right from your chat interface without needing console access.

Data Team Lead

You get quick, high-level overviews of the health across multiple environments to report status to stakeholders fast.

What Changes When You Connect

01 Quickly check job status: Instead of navigating logs, you simply ask your agent to check the run progress. This saves minutes on every debugging session.

02 Targeted execution: You don't need full UI access to start a pipeline. By using tools like `trigger_job`, you can isolate and test specific transformations directly through conversation.

03 Full visibility: With one call, your agent runs `list_environments`, giving you an instant overview of every setup in your organization—a massive time saver for data leads.

04 Debug from chat: If a run fails, your agent doesn't just say 'fail.' It checks the status and helps retrieve job details, letting you debug complex failures instantly.

05 Better process control: You manage transformations via natural language. This keeps your entire workflow history centralized with your AI client, not scattered across multiple tabs.

Real-World Applications

Needing to check if Production data is ready for launch

A team lead needs confirmation that the latest dataset finished running correctly. They ask their agent: 'What's the status of the production pipeline?' The agent runs `get_run_status` and confirms, 'The run is at 98%, all fact nodes materialized.' This prevents last-minute deployment errors.

Debugging a failed Staging environment transformation

An analytics engineer notices that staging data looks wrong. They ask their agent to check the details using `get_job_details`. The agent reports which specific nodes failed and why, letting them fix it immediately without opening any dashboards.

Testing a new pipeline segment before full deployment

A data engineer wants to test a small change on the Dev environment. They ask their agent to `trigger_job` for that specific job, ensuring they don't accidentally run it against Production or waste compute resources.

Getting an overview of all possible environments

A new team member needs to know where the data lives. They ask their agent to `list_environments`, which instantly returns a list of 'Dev,' 'Staging,' and 'Production' with status indicators, getting them up to speed in seconds.

Patterns to Avoid

Writing raw API calls

✗ AVOID

The user manually constructs a JSON payload for `trigger_run` or tries to manage environment lists using basic HTTP requests. This is slow, error-prone, and requires knowing exact endpoints.

✓ INSTEAD

Instead of writing code, ask your agent: 'Trigger the main pipeline run in Staging.' Your MCP translates that natural request into the correct tool call (`trigger_run`) automatically.

Confusing environments with jobs

✗ AVOID

A user asks to check a job status but specifies an environment name. The system might fail because it needs specific run IDs, not just names.

✓ INSTEAD

To check the status accurately, first ask your agent to `list_environments` to confirm the correct ID, then use that ID with `get_run_status`. Always verify context first.

Overlooking available nodes

✗ AVOID

A user only knows they need a transformation but doesn't know which specific part of the pipeline needs updating.

✓ INSTEAD

Use the `list_nodes` tool to retrieve metadata about all available nodes in that environment first. Then, guide your agent: 'Focus on node X and check its details.'

The Right Fit

Use this MCP if your job involves frequently monitoring or triggering complex data transformations within Snowflake environments. Specifically, if you need to list multiple environments (`list_environments`), check the status of different runs (`get_run_status`), or initiate specific jobs without opening a graphical interface, this is for you. Don't use it if your only goal is simple querying of static metadata; there are simpler search tools for that. If you just need to view historical data records and don't care about the pipeline health, you might be better off with a direct database query MCP.

Coalesce MCP: Managing Snowflake Pipeline Failures via AI Agents

Today, checking on data pipelines is painful. You have to open the Coalesce UI, navigate to the correct environment, find the job run history, and then manually check if it failed or just stalled out. This means constantly switching context, copy-pasting IDs, and guessing which dashboard shows the right thing.

With this MCP, you stop clicking. You tell your agent: 'Why did the Production pipeline fail?' The agent handles all the logic; it checks the job status, retrieves the error logs using `get_job_details`, and gives you a clear answer in chat. It's immediate diagnosis.

Coalesce MCP: Running Transformations on Demand for Snowflake Data

Manually, triggering a test run requires finding the exact job name and ensuring you have selected the correct node selector. If you miss one step, or target the wrong environment, you waste compute credits and delay your work.

Now, you simply ask your agent to trigger the run. You state the goal—'Run the core ETL jobs for Staging.' The MCP takes care of selecting the right tools (`trigger_run` or `trigger_job`) and initiating the pipeline immediately. It’s reliable.

8 Coalesce Tools for Data Pipeline Control & Job Monitoring

Use these tools to list environments, check run statuses, trigger jobs, and pull detailed information about any data transformation job in your Snowflake warehouse.

#	TOOL	DESCRIPTION
01	`get_environment`	Retrieves detailed configuration information for a specific data pipeline environment.
02	`get_job_details`	Fetches comprehensive details about a particular job, including its historical performance and settings.
03	`get_run_status`	Checks the current progress or final status of any triggered data pipeline run.
04	`list_environments`	Retrieves a comprehensive list of all environments configured in your Coalesce organization.
05	`list_jobs`	Gets a roster of available jobs, with the option to filter them by which environment they belong to.
06	`list_nodes`	Retrieves metadata about specific transformation nodes within an active environment.
07	`trigger_job`	Manually starts a predefined data transformation job inside a specified environment.
08	`trigger_run`	Initiates a brand new, full run for an entire environment, optionally targeting a specific job within it.

See It in Action

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

U I need a status update on my Production data warehouse pipelines.

Production Pipeline Status Report

- **Environment:** Production (env-001)
- **Last Run Status:** ✅ Success, 2 hours ago
- **Current Activity:** All core dimension tables refreshed successfully.
- **Next Steps:** Fact nodes are currently materializing in Snowflake. Estimated completion: 30 minutes.

U List all my data environments and tell me which one failed recently.

You have three configured environments:
1. 'Production' (env-001) — Last run: success, 2h ago
2. 'Staging' (env-002) — Last run: success, 5h ago
3. 'Development' (env-003) — **Last run: failed**, 1 hour ago.

Do you want me to retrieve the detailed error logs for Development?

U Start a test run in Staging using only the sales node.

Okay, initiating job 'sales-etl' in the Staging environment. Run ID: `run-8k3m`. Status: Running.

- **Target:** Sales Node Metadata retrieved successfully.
- **Progress:** 10% complete (Initializing connection).
- **Expected Time:** Approximately 4 minutes.

Frequently Asked Questions

01 **How does the Coalesce MCP help me monitor job status?**

The Coalesce MCP lets your agent check pipeline progress instantly. You can ask for a current run status, and it tells you the percentage complete or if there was an error, saving you from manually checking dashboards.

02 **Can I use Coalesce MCP to start new data pipelines?**

Yes. You can trigger jobs and full pipeline runs on demand. This means when a test is needed or an urgent update hits, your agent starts the transformation for you without needing UI access.

03 **Does Coalesce MCP work with my existing Snowflake setup?**

Absolutely. Because it connects directly to the Coalesce platform built on Snowflake, it manages transformations and data pipelines exactly where your data lives, making everything cohesive.

04 **What information does the Coalesce MCP give about environments?**

It gives you a full picture. You can list all configured environments—Dev, Staging, Prod—and pull specific details for any one environment to verify its setup parameters.

05 **What if I need to debug a failed run using Coalesce MCP?**

You just ask your agent. It can check the job's history and retrieve detailed logs, pointing out exactly which step or node caused the failure. This cuts down debugging time from hours to minutes.

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 `"coalesce": { "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

Coalesce 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 Coalesce. 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	Coalesce MCP
Server ID	`019d7575-7acf-730a-aeac-50fd1ce74166`
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/coalesce.