

MCP SERVER

NO CODE

CLOUD HOSTED

# Confluent MCP for AI Agents

Govern Kafka clusters and topic data across environments

Confluent MCP gives your AI agent direct access to the Confluent Cloud API, letting you manage complex Kafka clusters and data streams through natural language commands. You can check cluster health, list topics across environments, audit connectors, and verify configuration states without logging into a dashboard.

**A+** Quality Score 100/100

apache-kafka

data-streaming

event-processing

cluster-monitoring

topic-management



# 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.

### 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.

### 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.

### 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**

Encryption at rest

**Ed25519**

PKI vault signatures

**24h TTL**

Ephemeral session keys

**V8 Isolate**

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.

### 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.

### 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.

### 06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

**FinOps**

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

**Firewall & DLP**

PII redaction activity, sensitive data protection counters, and security event timeline.

**Agent Activity**

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

**Tool Health**

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

**Incident Log**

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

Get started at [cloud.vinkius.com](https://cloud.vinkius.com) — connect your AI agent in under 60 seconds.

# Confluent MCP

7 tools available

Cloud-hosted on Vinkius

This MCP connects your AI client directly to Confluent Cloud, the platform built on Apache Kafka for enterprise data streaming. It lets you treat your entire streaming infrastructure—from clusters to individual topics—as if it were sitting right in your chat window. Instead of navigating complex dashboards or writing boilerplate CLI scripts, you simply ask your agent what you need. For instance, you can request a list of all active environments, then check the health of a specific cluster within that environment, and finally audit which connectors are running. It's about doing deep infrastructure work using just conversation. If you're building out an advanced AI toolset, Vinkius makes it easy to connect this level of data governance capability into your existing workflow.

---

## Core Capabilities

### 01 — Audit and list all environments

Discover the necessary environment IDs needed to perform operations on specific clusters or connectors.

### 03 — Check cluster health metrics

Get detailed information on a specific cluster's configuration, endpoint URLs, availability, and provisioning status.

### 05 — Monitor data pipelines (connectors)

View the status of configured source and sink connectors, confirming if your data ingestion pipes are running correctly.

### 02 — Inventory Kafka clusters

Retrieve a comprehensive list of all your available Kafka clusters, including their cloud provider and region status.

### 04 — Manage streaming topics

List all existing Kafka topics, viewing their partition count and replication configurations at a glance.

### 06 — Audit programmatic access

List service accounts or retrieve Cloud API keys to audit who has access to your Kafka organization.

# One Click on Vinkius — From Prompt to Execution

Available at [vinkius.com/mcp/confluent](https://vinkius.com/mcp/confluent) — connect your AI agent in three steps.

- 01 Add the Confluent integration to your AI client's toolset.
- 02 Provide your necessary Confluent Cloud API Key and Secret credentials.
- 03 Ask your agent a natural language question, like 'What is the CPU utilization for the main production cluster?'

The bottom line is: you use simple conversation to perform complex data governance tasks that usually require multiple clicks in a specialized dashboard.

---

## Built For

This MCP is built for the people who manage the pipelines. If your job involves knowing the state of 10+ Kafka topics or verifying if a cluster endpoint changed overnight, this tool saves you from context switching between dashboards and CLIs.

### Data Engineer

You use this to verify pipeline readiness by checking topic configurations (like partition counts) before deployment, or listing connectors to confirm data sources are active.

### DevOps Team Lead

You monitor cluster health and review environment configurations across multiple regions without having to manually open the Confluent dashboard every time.

### System Architect

You audit stream definitions, check service accounts, and map out which environments are connected before planning a major new data integration.

## What Changes When You Connect

- 
- 01 Quickly check cluster health using `get_cluster_details` . Instead of SSHing into a machine to verify node status, your agent gives you an immediate summary of CPU metrics and availability.

---

  - 02 Audit entire stream definitions by calling `list_environments` first. This ensures you have the correct IDs before attempting to inspect topics or connectors in a specific scope.

---

  - 03 Verify data pipeline integrity by running `list_connectors` . You can instantly confirm if your critical source-sink connections are active and report any failed tasks.

---

  - 04 Maintain security visibility using `list_service_accounts` and `list_cloud_api_keys` . You get an immediate, centralized audit log of who has access to the core Kafka infrastructure.

---

  - 05 Streamline topic governance with `list_topics` . See every relevant stream's partition count and replication factor without running multiple manual commands.

---

  - 06 Reduce operational friction by combining checks. Your agent can list environments, then check clusters in that environment, all from one prompt.
- 

---

## Real-World Applications

### A cluster is suddenly showing high latency

The DevOps team needs to know the root cause quickly. They ask their agent to run `get_cluster_details` and check the status of all connectors via `list_connectors` . The response immediately flags a specific connector as failing, directing the team straight to the broken pipeline.

### Pre-deployment topic audit is required

A Data Engineer needs to know if their new topic setup meets governance standards. They use `list_topics` to check existing topics for correct partition counts and replication factors across all environments, verifying compliance before committing the change.

### Investigating unauthorized access

An architect suspects a breach. They run ``list_service_accounts`` and immediately review the list of authorized credentials alongside using ``list_cloud_api_keys`` to see if any keys were recently generated outside normal process.

### Mapping out all data sources

A new team member needs a full picture of the streaming landscape. They ask their agent to first run ``list_environments``, then use that ID list to check every active cluster using ``list_clusters``.

---

## Patterns to Avoid

---

### Treating Kafka like simple file storage

#### X AVOID

A user asks the agent to simply 'check all data.' This is too vague and doesn't specify which environment or cluster, leading to an unmanageable dump of irrelevant information.

#### ✓ INSTEAD

Always scope your request. Instead of general checks, ask for specific actions: use ``list_environments`` first, then narrow the query by asking for 'all topics in [specific environment ID]' using ``list_topics``.

### Ignoring cluster dependencies

#### X AVOID

A user runs a topic check without knowing which environment it belongs to. The agent might return stale data or fail entirely because the necessary context (environment ID) was missing.

#### ✓ INSTEAD

Always start by calling ``list_environments`` to establish scope. This ensures your subsequent calls, like listing clusters via ``list_clusters``, are correctly scoped.

### Relying on outdated credentials

#### X AVOID

A user tries to run a connector status check using old API keys that have been revoked or rotated out of the system.

#### ✓ INSTEAD

Before running any critical checks, audit your access by calling ``list_service_accounts`` and checking ``list_cloud_api_keys``. This confirms you are working with current credentials.

---

## The Right Fit

Use this MCP when the core problem is operational visibility into a complex, multi-environment Kafka setup. If your workflow requires knowing *what* topics exist, *where* they live (environments), or *if* a cluster's health deviates from baseline—this is essential. However, don't use it if you just need basic data querying; this isn't a search

engine. For instance, if you only need to read the contents of a single record, that's outside its scope. This MCP manages metadata and state. If your task involves provisioning entirely new infrastructure from scratch, look for dedicated CI/CD tools instead.

---

## Confluent MCP: Auditing Kafka Stream Infrastructure with Confluent

Before this MCP, auditing a streaming data platform was an exercise in context switching. You'd log into the web dashboard to find environment IDs, then switch to the CLI to list clusters and their details, followed by another session just to check topic metadata. It's manual, time-consuming, and prone to missing an entire environment.

Now, your agent handles all of that complexity. You simply ask: 'Show me the status of my production Kafka stream infrastructure.' The MCP orchestrates calls like `list_environments` and then uses those IDs to run comprehensive checks on clusters and topics, giving you a single, cohesive report.

---

## Confluent MCP: Monitoring Data Flow Reliability with Confluent

The biggest time sink used to be verifying data flow. Did the connector actually move the data? Was it running against the right cluster? Engineers spent hours cross-referencing logs and dashboards just to confirm basic operational status.

With this MCP, you check reliability in seconds. By calling `list_connectors` or using `get_cluster_details`, your agent confirms not only that the connection exists, but that it's actively processing data according to your query.

---

## 7 Confluent Tools for Streaming Data Governance

Use these tools to get specific details on clusters, topics, connectors, and service accounts across your entire streaming infrastructure.

#	TOOL	DESCRIPTION
01	<code>get_cluster_details</code>	Retrieves detailed information on a specific Kafka cluster, including its endpoint URLs and availability status.
02	<code>list_cloud_api_keys</code>	Gets a list of all API keys currently active in your Confluent Cloud account for security auditing.
03	<code>list_clusters</code>	Returns an overview of all Kafka clusters in your organization, showing their status and cloud provider region.
04	<code>list_connectors</code>	Retrieves the configured source and sink connectors for a given environment or cluster to check their operational status.
05	<code>list_environments</code>	Provides a list of available Confluent Cloud environments, which are needed identifiers for other operations.
06	<code>list_service_accounts</code>	Retrieves service accounts used in your organization, useful when auditing programmatic access permissions.
07	<code>list_topics</code>	Returns a comprehensive list of all topics, including their partition count and replication settings for quick review.

---

## See It in Action

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

**U** Which environments do I have available for my Kafka streams?



### Confluent Cloud Environments Found:

- `staging-us` : Used for testing new features.
- `production-eu` : Live environment for European data.
- `dev-sandbox` : Sandbox for individual developer work.

You can use these IDs to check specific cluster details or topic lists.

**U** Check the status of all my Kafka clusters and tell me which one is in the EU region.



I found 3 active clusters:

- **Cluster A (US-EAST)**: Status: Healthy. Provisioning: Complete.
- **Cluster B (EU-WEST)**: Status: Warning. Node count mismatch detected on node 4.
- **Cluster C (US-WEST)**: Status: Healthy. Provisioning: Complete.

The EU cluster ( `Cluster B` ) needs attention regarding its node count.

**U** List all topics that are part of the staging environment and tell me their partition counts.



### Topics in `staging-us` :

TOPIC NAME	PARTITIONS	REPLICATION FACTOR
user_login_events	12	3
payment_updates	6	3
analytics_stream	8	3

These topics look correctly configured for staging.

---

# Frequently Asked Questions

---

**01 How can I check the current health of all my Kafka clusters with Confluent MCP?**

You can get a list of all clusters, then request detailed status reports for any specific one. This tells you if nodes are available and provides key metrics like CPU usage.

---

**02 I need to see every active topic in my streaming platform.**

Confluent MCP allows you to list all topics across a given cluster, providing the partition count and replication factor for quick compliance checks.

---

**03 How do I use Confluent MCP to audit data access?**

You can audit access by listing service accounts or retrieving API keys. This gives you a clear record of who has programmatic rights across your Kafka organization.

---

**04 Does Confluent MCP help me find the IDs for different environments?**

Yes, it provides a list of all available environments. Knowing these IDs is critical because they are needed to scope any cluster or connector operation correctly.







---

# Go Live in 60 Seconds

Get your connection token from [cloud.vinkius.com](https://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
 <b>Claude AI</b>	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
 <b>Cursor</b>	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
 <b>VS Code</b>	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add <code>"confluent": { "url": "..." }</code>
 <b>Windsurf</b>	MCP Settings → <code>mcp_settings.json</code> → Add endpoint URL
 <b>ChatGPT</b>	Settings → Tools & plugins → Add MCP server → Paste endpoint
 <b>Gemini</b>	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

# Confluent 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](https://cloud.vinkius.com) →

[vinkius.com](https://vinkius.com) · [support@vinkius.com](mailto: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 Confluent. 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	Confluent MCP
Server ID	019d757a-1485-703a-aa93-dc6683841eab
Platform	Vinkius Cloud for AI Agents
Endpoint	<a href="https://edge.vinkius.com/{token}/mcp">https://edge.vinkius.com/{token}/mcp</a>

### 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/confluent](https://vinkius.com/mcp/confluent).