

MCP SERVER

NO CODE

CLOUD HOSTED

Weaviate MCP for AI Agents

Semantic Search and Vector Data Management for Knowledge Graphs

Weaviate connects your AI agents directly to a vector database, allowing you to manage complex semantic search tasks through plain conversation. Instead of writing complicated queries or navigating raw console interfaces, your agent performs nearest neighbor similarity searches across massive collections of data. You can audit schemas, discover objects by UUIDs, and even check the operational health of the entire cluster using only natural language commands.

A+ Quality Score 100/100

semantic-search

vector-embeddings

schema-management

nearest-neighbor

ai-native

data-indexing



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 — connect your AI agent in under 60 seconds.

Weaviate MCP

7 tools available

Cloud-hosted on Vinkius

Weaviate lets you use vector search and semantic data management without getting bogged down in database syntax. Your AI agent handles complex operations that used to require manual JSON querying inside a specialized console. You just ask your agent to find things, and it performs the necessary searches against your structured, vectorized knowledge base.

When you connect this MCP via Vinkius, your preferred AI client gets full administrative access to your vector data—it acts as if it's an integrated database administrator. It helps developers test semantic search queries for relevance, allows data engineers to audit schemas across collections, and gives SRE teams a straightforward way to monitor cluster health status. The result is that you can surface relevant documents or analyze structured object metadata using nothing but conversation.

Core Capabilities

01 — Perform Semantic Vector Searches

The MCP performs nearest neighbor similarity searches across specific collections, finding content based on meaning rather than just keywords.

03 — List and Discover Data Objects

The agent supports browsing collections to list objects, providing full property values and vector metadata for auditing purposes.

05 — Monitor Cluster Operations

The MCP provides operational status, node health reports, and resource usage details for the entire Weaviate cluster.

02 — Audit Database Structure

You can retrieve the definitions for any specific data collection or pull the complete schema definition for all available classes in your instance.

04 — Inspect Specific Records

Retrieve detailed data object information using a unique identifier (UUID) to check specific metadata or internal configurations.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP and enter your required Weaviate Base URL and API Key.
- 02 Connect your credentials to any MCP-compatible client, like Cursor or Claude. Your agent now has access to the database endpoints.
- 03 Ask your AI client a question (e.g., 'Show me the schema for the Product class') and let it handle the data retrieval and formatting.

The bottom line is that your AI agent translates natural language requests into complex vector operations, eliminating the need to write database query languages manually.

Built For

This MCP targets technical teams who deal with massive amounts of indexed data. It's perfect for Data Engineers frustrated by manual schema auditing, AI Developers needing quick vector search validation, and DevOps staff who need real-time cluster status without logging into a dedicated console.

Data Engineer

Uses the MCP to audit database schemas across multiple collections or browses indexed objects to verify data consistency.

AI Developer

Tests and refines semantic search queries, ensuring their vector ingestion pipelines are accurate before deployment.

SRE / DevOps Engineer

Monitors the operational health of Weaviate nodes and manages instance configuration details to maintain system stability.

What Changes When You Connect

- 01 Audit your entire data structure using `get_full_schema` without writing complex queries. You get a single, comprehensive overview of all classes in one go.

-
- 02 Quickly assess cluster stability with `get_cluster_nodes` . Your agent reports operational health and resource usage metrics instantly, saving manual dashboard checks.
 - 03 Deep dive into specific records using `get_object_details` by UUID. This lets you inspect metadata and internal configurations for compliance or debugging purposes.
 - 04 Refine your search logic immediately. Use the `search_near_vector` tool to test semantic relevance with just a query vector, validating results before production deployment.
 - 05 Understand data limitations using `get_class_schema` . You can confirm exactly which properties exist in any given collection before building an AI workflow around it.
-

Real-World Applications

Troubleshooting a Failed Search Query

A developer needs to know if the 'Article' class schema is configured correctly. Instead of guessing, they ask their agent and use `'get_class_schema'` to confirm property types and required fields before rewriting their search logic.

Validating Model Performance

An AI developer builds a new retrieval model. They use the agent to execute multiple `'search_near_vector'` queries with known vectors to measure similarity scores and ensure the system is retrieving conceptually relevant content.

Checking Data Integrity on a Live System

A data engineer suspects some records are missing crucial metadata. They can use `'list_objects'` combined with `'get_object_details'` to quickly sample and audit the properties of several objects, confirming if the expected fields are present.

On-Call System Monitoring

An SRE gets an alert about high resource usage. They immediately ask their agent to run `'get_cluster_nodes'` to see which specific node is maxing out CPU, getting actionable data faster than navigating the dashboard.

Patterns to Avoid

Treating Weaviate like a simple SQL database

✗ AVOID

Attempting to retrieve complex relationships or perform joins by writing basic SELECT statements. This ignores the vector nature of the data.

✓ INSTEAD

Always use ``search_near_vector`` when you need conceptual matching, not exact field matches. If you are auditing structure, start with ``get_full_schema`` to understand what's possible.

Ignoring operational status

✗ AVOID

Assuming the cluster is healthy because the main dashboard looks green, but failing when a specific node fails under load.

✓ INSTEAD

Before relying on search results, run ``get_cluster_nodes``. This verifies that all components are running optimally and gives you resource usage metrics.

Asking for generalized data

✗ AVOID

Simply asking the agent to 'show me all user data' without knowing which class holds it, leading to ambiguous or incomplete results.

✓ INSTEAD

First run ``get_full_schema`` to list available collections. Then use ``list_objects`` on the correct class name to narrow down your scope.

The Right Fit

Use this MCP if your primary need is semantic search, vector embedding management, or deep database auditing capabilities. Specifically, it's essential when you must perform nearest neighbor lookups that go beyond simple keyword matching. Don't use it if you are building a purely transactional system (like managing user accounts where only CRUD operations matter). For pure data logging and tracking state changes without complex querying, a simpler key-value store might suffice. However, if your workflow requires understanding the *meaning* behind vast amounts of stored information or validating the health of a large distributed cluster, this MCP is non-negotiable.

Weaviate: Mastering Semantic Search and Vector Data Management

Today, finding relevant information in massive knowledge graphs means manually querying dozens of collections via specialized console UIs. You're clicking through schemas, copying UUIDs, and running multiple `list_objects` commands just to confirm if the data structure supports your search intent.

With Weaviate connected through this MCP, you tell your agent what you need—say, 'Find all articles related to machine learning in 2023.' The agent executes a complex vector similarity query using `search_near_vector` and returns only the top results, saving hours of manual investigation.

Weaviate: Operational Oversight for Vector Databases

Before deploying any new search feature, you have to verify the underlying database health. This means jumping between monitoring dashboards and running separate schema checks just to confirm everything is operational.

Now, your agent handles it all. You can ask for a cluster status check using `get_cluster_nodes` or audit metadata with `get_instance_metadata`. It consolidates complex infrastructure information into simple, actionable reports.



Weaviate: 7 Tools for Vector Search and Schema Management

These tools let your AI agent interact directly with the database structure, allowing you to list objects, check metadata, or perform complex nearest neighbor vector lookups.

#	TOOL	DESCRIPTION
01	get_class_schema	Retrieves a detailed definition for a specific data collection (class) within the instance.
02	get_instance_metadata	Pulls high-level configuration details, including the server version and enabled modules.
03	get_cluster_nodes	Provides operational metrics on node status and resource usage across the entire cluster.
04	get_object_details	Fetches a specific data object's metadata by providing its unique UUID for deep inspection.
05	get_full_schema	Retrieves the complete schema definition, showing all classes available in the entire Weaviate instance.
06	list_objects	Lists data objects within a specific class, supporting basic pagination limits for browsing.
07	search_near_vector	Executes a nearest neighbor vector search by providing a target class name and a query vector array.

See It in Action

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

U List all classes in my Weaviate schema.



I found 3 classes in your schema: 'Article', 'Product', and 'UserFeedback'. Would you like to see the property definitions for any of these classes?

U Search the 'Article' class for items similar to this vector: [0.12, -0.05, 0.88, ...].



I found 5 similar articles. The top match is 'Vector Search for Beginners' (UUID: 550e8400-e29b...) with a similarity score of 0.94. Would you like to see the full content of this article?

U What is the current health status of my Weaviate cluster nodes?



All 3 nodes in your cluster are currently 'HEALTHY'. Node-1 is using 42% CPU and 6.5GB RAM. Node-2 and Node-3 are also within normal operating parameters. No issues detected.

Frequently Asked Questions

01 How do I use Weaviate MCP for AI Agents to find related documents?

You ask your agent a question in plain English. It handles the complexity of vector search, finding results based on meaning and context rather than just matching keywords. This is perfect for research or content discovery.

02 Can Weaviate MCP for AI Agents help me check my database structure?

Yes, you can ask the agent to pull the complete schema definition or look up a specific collection's properties. This is useful for data engineers who need to audit what fields are available in your data set.

03 If I run into database errors, how does Weaviate MCP for AI Agents help me?

You can ask the agent about the cluster's operational health. It reports on node status and resource usage, giving you immediate insight into whether a technical failure is due to capacity or configuration.

04 What if I need to see details for one specific record in Weaviate MCP for AI Agents?

You just provide the unique identifier (UUID) to your agent. It then executes a deep audit, giving you all associated metadata and configuration details for that single object.

05 Is this better than just using simple database search tools?

Yes. Simple databases are good at exact matches (like finding 'John Doe'). Weaviate uses vectors to find things by **concept**—it finds results that mean the same thing, even if they use different words.

06 Can I perform a vector search using float arrays through the agent?

Yes. The ``search_near_vector`` tool allows you to perform semantic searches by providing a query vector as a JSON array of floats. Your AI agent will return the most similar objects from your Weaviate collection.

07 How do I see the data structure of my Weaviate collections?

You can use the ``get_full_schema`` tool to see all classes and properties defined in your instance, or ``get_class_schema`` if you want to focus on a specific collection's definition.

08 Is it possible to monitor cluster health via chat?

Absolutely. Use the ``get_cluster_nodes`` tool to retrieve operational data for all nodes in your Weaviate cluster, including their current status and resource utilization metrics.

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 <code>"weaviate": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Weaviate 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 Weaviate. 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	July 2026
MCP Server	Weaviate MCP
Server ID	019d7620-f2dc-7299-87a2-a2c2f227ec79
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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/weaviate.