MCP SERVER | NO CODE | CLOUD HOSTED

ClickHouse (Vector Search) MCP for AI Agents

Query Vector Embeddings and Execute Complex OLAP Queries

ClickHouse (Vector Search) provides AI agents with direct access to your massive analytical database. It lets you manage vector embeddings and run complex SQL queries conversationally, performing high-speed semantic searches without writing boilerplate code.

A+ Quality Score 100/100

olap | vector-embeddings | sql-execution | high-performance-data | real-time-analytics

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/clickhouse-vector-search

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.

ClickHouse (Vector Search) MCP

7 tools available
Cloud-hosted on Vinkius

Managing modern data means juggling structured records, specialized metadata, and high-dimensional vectors. This MCP connects any compatible AI client directly to your ClickHouse cluster, letting your agent speak the language of your database. Instead of navigating dozens of tabs or writing complex connection strings, you simply ask questions in natural English.

Your agent handles everything from listing available schemas and running arbitrary DML or SELECT statements to identifying mathematical distance traces using advanced vector metrics like cosineDistance. You can audit cluster health—checking row counts and compression ratios—or verify exact capability branches, such as HNSW support. By connecting your ClickHouse data through Vinkius, you get full control of both your raw SQL records and your semantic search capabilities from one central point.

Core Capabilities

01 — View database schemas

List all available logical databases and inspect the detailed structure of tables within them.

02 — Execute structured queries

Run any complex SQL query, including data definitions (DDL), data manipulation (DML), or simple SELECT statements.

03 — Perform vector similarity searches

Find the closest matching records in your dataset by calculating mathematical distances between vector embeddings.

04 — Analyze table health statistics

Pull internal structural states, including row counts and compression ratios, to audit how healthy a specific table is.

05 — Check cluster configuration

Retrieve the version number and binary limits of the ClickHouse instance to verify its current capability set.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/clickhouse-vector-search** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your ClickHouse URL, along with your username and password.

02 Your AI client authenticates and gains read/write access to the specified database cluster.

03 You prompt your agent in natural language; it translates that intent into specific SQL or vector operations against your data.

The bottom line is, you get to run complex analytical tasks—from simple reporting to advanced semantic matching—using only conversational prompts.

Built For

This MCP targets technical roles that spend time translating business questions into database queries. It's for the data analyst stuck in manual report generation, the developer needing rapid vector testing, and the DBA who needs constant oversight of cluster performance.

Data Analyst

Runs ad-hoc reports across multiple tables by asking natural language questions instead of writing complex JOIN statements.

AI Developer

Tests and fine-tunes vector similarity search pipelines rapidly, debugging semantic matching without needing to write boilerplate data access code.

Database Administrator

Monitors the operational health of large clusters by querying metrics like compression ratios and instance versions on demand.

What Changes When You Connect

01 Stop writing complex SQL. You tell your agent what data you need, and it generates the necessary query using `execute_sql`.

02 Analyze vector similarity without coding boilerplate. Use `vector_search` to find semantic matches just by describing the relationship you're looking for.

03 Maintain cluster health easily. Run `get_table_stats` to instantly check row counts and compression ratios across multiple tables.

04 Understand your data model quickly. Use `describe_table` to get a deep, reliable schema inspection without needing to consult documentation.

05 Verify system limits on the fly. Check capability branches using `get_version` to confirm support for features like HNSW.

Real-World Applications

Auditing data quality after a migration

A DBA needs to check if the new 'sales' table has been loaded correctly and if compression rates are within acceptable limits. They use `get_table_stats` and then `get_version` to confirm both structural integrity and platform capability.

Building a document search engine

An AI developer needs to find documents semantically related to 'supply chain risk' using vectors. They prompt their agent, which executes the `vector_search` tool, delivering the top matches and associated metadata immediately.

Generating a quarterly business review report

A data analyst needs to combine sales figures (DDL/DML) with user behavior metrics across three different tables. Instead of writing five separate queries, they ask their agent to run the necessary `execute_sql` commands and compile the results.

Prototyping a new data source integration

A product team wants to know if a specific column type (like Array(Float32)) is available in an existing database. They use `describe_table` to inspect the schema, confirming feasibility before writing any code.

Patterns to Avoid

Manually checking table structure

✗ AVOID

A developer spends 20 minutes clicking through multiple UI tabs and running `SHOW COLUMNS` commands just to verify if a column supports vector types.

✓ INSTEAD

Instead, let your agent run the `describe_table` tool. This provides an immediate, comprehensive schema extraction in one step, confirming data properties instantly.

Treating all data queries as pure SQL

✗ AVOID

A user tries to find similar documents by writing a SELECT query that only filters on text keywords, missing the advanced semantic similarity search capability.

✓ INSTEAD

For finding conceptual matches (like 'best practices for energy efficiency'), use the `vector_search` tool. This compares vector embeddings, which is far more accurate than simple keyword matching.

Ignoring cluster operational limits

✗ AVOID

A DBA assumes that a new feature requiring advanced indexing (like HNSW) works because it worked last year, but the current instance version doesn't support it.

✓ INSTEAD

Always run `get_version`. This tool checks the precise active cluster limits and confirms if the necessary binary functionality is available for your specific deployment.

The Right Fit

Use this MCP if your workflow requires connecting natural language conversation directly to high-performance analytical data, especially when you deal with vector embeddings or need to run complex SQL. It’s perfect for developers and analysts who spend more time asking 'what if' questions than writing boilerplate code.

Don't use it if all you need is a simple key-value lookup on a single field (a standard API call handles that better). Also, don't rely on this MCP to replace data governance; while `list_databases` helps map schemas, you still need human sign-off before making changes with `execute_sql`. If your primary goal is just basic CRUD operations and not analytics or vectors, a simpler database connector might suffice.

ClickHouse (Vector Search) MCP: Automating Data Exploration in OLAP

Right now, getting a full picture of your data requires a painful cycle of investigation. You have to run one query just to list the available databases, then another for every table you suspect has relevant data. If you need to know what specialized types are available—like those tricky Array(Float32) vectors—you're forced to manually check schemas and read documentation pages.

With this MCP, your agent takes over that investigative process. You simply ask, 'What databases do I have?' or 'Show me the schema for the user table.' The system handles the underlying calls—like `list_databases` and `describe_table`—and presents you with a cohesive, actionable answer. You get instant data context.

ClickHouse (Vector Search) MCP: Managing High-Dimensional Analytics

Building semantic search usually means writing multiple components: an embedding generator, a

Using this MCP, you bypass that entire development stack. You tell your agent to 'Find

database connector, and the query logic itself. This requires managing dozens of lines of code just to perform one similarity check.

documents similar to X,' and it executes `vector_search`. You get results with actual mathematical distances (like cosineDistance) returned immediately, making advanced data analysis accessible without writing a single line of vector math.

7 Tools in ClickHouse (Vector Search) for Data Analytics

Use these tools to manage schemas, execute SQL statements, perform high-speed vector matching, and monitor database performance metrics.

#	TOOL	DESCRIPTION
01	`list_databases`	Identifies all top-level schemas available within your ClickHouse cluster.
02	`list_tables`	Retrieves the list of specific tables housed inside a selected database.
03	`describe_table`	Provides a detailed structural breakdown, including all column types and properties for a chosen table.
04	`execute_sql`	Runs any arbitrary SQL query (SELECT, INSERT, UPDATE, DELETE) directly against the cluster.
05	`vector_search`	Calculates and identifies mathematical distances between vector embeddings to find semantic matches.
06	`get_table_stats`	Extracts key internal metrics, such as row counts and compression ratios, for a given table.
07	`get_version`	Retrieves the precise version information and binary limits of the ClickHouse execution instance.

See It in Action

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

U What are the total row counts for my top three sales tables?

Table Health Report: Sales Data

TABLE NAME	ROWS COUNTED	COMPRESSION RATIO
`analytics_prod.sales`	1,200,000	×4.2
`analytics_prod.leads`	85,000	×3.1
`system.metrics`	9,100	×6.0

Note: All counts are current as of the last successful query.

U Find me records that talk about 'sustainable energy' in the embeddings table.

Vector Search Complete

Found 5 matches for 'sustainable energy'. The top hit is from `analytics_prod.documents`.

- **Document ID:** D401
- **Distance Score (cosine):** 0.032
- **Snippet:** "...requires shifting focus toward sustainable energy sources by 2035."

Would you like to see the associated metadata for this document?

U I need a list of all available databases and what columns they contain.

Database Schema Overview

We found four primary schemas:

1. `default` : Contains general system tables.
2. `analytics_prod` : Primary data lake for reports (e.g., `sales` , `users`).
3. `vector_store` : Dedicated schema holding high-dimensional embeddings.
4. `system` : Operational metrics and metadata.

For example, the `analytics_prod` schema includes tables like `sales` which has columns for `sale_id (Int64)` , `date (Date)` , and `vector_data (Array(Float32))` .

Frequently Asked Questions

01 **How do I perform advanced semantic searches using the ClickHouse MCP?**

You can run high-speed vector similarity searches by telling your agent what concept you are looking for. The MCP executes this search against your embeddings, returning records that match conceptually, not just keyword-for-keyword.

02 **Can the ClickHouse MCP help me write or fix SQL queries?**

Yes. You can use natural language to request complex reports or data modifications. The MCP translates your intent into precise SELECT, DML, or DDL statements that run directly against your cluster.

03 **What if I don’t know the schema of a table?**

No problem. You can ask the MCP to describe any table you point it toward. It immediately pulls up the column names, data types, and properties so you know exactly what data is available for querying.

04 **Is this ClickHouse (Vector Search) MCP good for monitoring database health?**

It's great for that. You can run commands to check the table stats, getting real-time metrics like row counts and compression ratios, which helps you audit performance without logging into a command line.

05 **Does this MCP support different types of data beyond simple text?**

It handles advanced analytical types, including specialized Array(Float32) vectors. This means you can build complex pipelines that combine traditional structured data with high-dimensional semantic information.

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 `"clickhouse-vector-search": { "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

ClickHouse (Vector Search) 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

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	ClickHouse (Vector Search) MCP
Server ID	`019d7572-4585-70e2-9194-0dbad1970531`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE