MCP SERVER | NO CODE | CLOUD HOSTED

ClickHouse MCP for AI Agents

Running lightning-fast OLAP Queries on Big Data

ClickHouse MCP connects your AI agent directly to an OLAP database for lightning-fast data analytics. You can run complex SELECT queries, check cluster replication status, and manage schemas using natural language instructions. It lets you query big data infrastructure without ever opening a SQL IDE.

A+ Quality Score 100/100

olap | sql | big-data | database-management | analytics

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

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 MCP

4 tools available
Cloud-hosted on Vinkius

Stop toggling between dashboards and databases just to get answers. This MCP connects your AI agent straight into ClickHouse, giving it the power to run complex analytical queries on massive datasets. You tell your agent what you need—like 'Show me Q3's sales by region'—and it executes the necessary reads using the `select_query` tool. Need to adjust a table? Use `execute_query` for schema changes and management actions. It even checks if your cluster is healthy with dedicated tools like `ping` or `replicas_status`. Getting connected is simple: just subscribe through Vinkius, input your connection details, and your agent does the heavy lifting. You get instant access to deep data insights without writing a single line of SQL.

Core Capabilities

01 — Execute a read-only query against the database

Runs SELECT statements to pull analytics and insights from the database without making any changes.

02 — Perform structural database modifications

Runs commands that change the database structure, like creating or altering tables.

03 — Verify the server's immediate network health

Checks if the entire ClickHouse server is online and responding to connection requests.

04 — Check data synchronization across replicas

Determines if data replicas across your cluster are lagging or synchronized correctly.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and provide your ClickHouse URL, username, and password.

02 Next, activate the connection within any compatible AI client, granting it read/write access permissions for the specified tasks.

03 Finally, prompt your agent with a natural language request—for instance, 'What was the average user count last month?'—and let it execute the query.

The bottom line is that you get to treat your data infrastructure like a conversation; you just talk to your AI client instead of writing code.

Built For

Anyone who needs reliable, high-speed answers from massive datasets. This MCP is built for the analyst stuck in endless dashboards and the data engineer who can't afford to switch contexts.

Data Analyst

Runs complex aggregations on raw logs or sales records without needing a dedicated SQL developer, getting instant summaries of large datasets.

Data Engineer

Quickly inspects table schemas and checks cluster replication status to verify data integrity during deployments or incident response.

DevOps Engineer

Monitors the overall health of the entire big data stack, verifying server availability with a single command before major maintenance windows.

What Changes When You Connect

01 Run deep analytics without writing SQL. The `select_query` tool handles complex aggregation, letting you instantly summarize massive datasets.

02 Maintain infrastructure health from one place. Use the `ping` and `replicas_status` tools to check cluster integrity before deployments.

03 Manage schemas directly through natural language. If a table needs updating, use `execute_query` to create or alter structures on the fly.

04 Avoid context switching entirely. Your agent keeps your data source connected whether you're running reports or fixing schema issues.

05 Optimize query performance by specifying resource limits. You can pass settings like max rows to read when using any SELECT capability.

Real-World Applications

Diagnosing a Data Pipeline Failure

A DevOps engineer notices replication lag and asks their agent for the cluster status. The agent uses `replicas_status` to immediately pinpoint which replica is failing, stopping downtime before users even notice.

Generating Quarterly Performance Reports

A data analyst needs Q3's sales summary across five different tables. Instead of writing a massive join query, they ask for the aggregate report; the agent uses `select_query` to pull all necessary metrics instantly.

Adding a New Tracking Field

A marketing team needs to track a new user ID. They prompt their agent to use `execute_query`, which creates the missing column in the main user table, updating the schema safely.

Quickly Validating Server Status

Before starting work on a big data project, an engineer simply asks if the database is available. The agent uses `ping` to confirm server health in seconds, preventing wasted time on offline systems.

Patterns to Avoid

Trying to change schemas with analytics queries

✗ AVOID

A user thinks they can use a SELECT query to force-add a column because the data is missing. This fails and corrupts nothing, but wastes time.

✓ INSTEAD

If you need to change structure, don't guess. You must explicitly ask your agent to run an altering command using `execute_query`.

Ignoring cluster health checks

✗ AVOID

Running a massive report that relies on data from all nodes without checking for replication lag first. The results will be incomplete or stale.

✓ INSTEAD

Always run the `replicas_status` tool first. It verifies that every node in your cluster is synced and ready before you pull any reports.

Overloading queries with resource limits

✗ AVOID

The agent runs a complex query without setting explicit limits, causing the database to consume too much memory and crash.

✓ INSTEAD

Always pass specific settings like `max_rows_to_read` or `max_execution_time` when running any read-only SELECT query.

The Right Fit

Use this MCP if your primary need is running advanced, high-volume analytical queries and managing large data schemas. You're dealing with OLAP reporting on gigabytes of logs or metrics. Don't use it if you just need simple CRUD operations (like updating a single record in a CRM) — for those, a dedicated API connector is better. If your main task is merely viewing basic metadata, you might only need the `select_query` tool. However, since this MCP also handles structural changes via `execute_query`, it's robust enough to cover both reporting and data governance.

Managing ClickHouse Schemas with AI Agents

Think about the process right now. You need to add a new column—say, 'IsVerified'—to your main user table because marketing started tracking it. That means logging into the database console, writing the `ALTER TABLE` command, making sure you specify the correct syntax and scope, hitting execute, and then manually confirming everything worked. It's slow, error-prone, and involves switching between multiple interfaces.

With this MCP, that whole process vanishes. You simply tell your agent: 'Add an IsVerified column to the users table.' The agent uses the `execute_query` action, handling the syntax and execution safely behind the scenes. You just get confirmation that the schema is updated.

Running Analytics Queries with ClickHouse MCP for AI Agents

Before this, running a big report meant manually writing complex joins across multiple tables and then running it. If you needed to tweak the parameters—like changing 'last 30 days' to 'last 60 days'—you had to copy-paste, edit the SQL, and run again. It’s tedious repetition.

Now, you just talk to your agent. You ask: 'Show me user engagement for the last two months.' The agent executes a read-only query using `select_query` and delivers the answer directly in the chat window. Your focus stays on the insight, not the syntax.

4 Tools for ClickHouse Analytics and Database Management

Use these tools to query read-only reports, execute structural changes, or monitor the health of your entire big data cluster.

#	TOOL	DESCRIPTION
01	`execute_query`	Runs commands that change the database structure, like creating or altering tables.
02	`ping`	Checks if the entire ClickHouse server is online and responding to connection requests.
03	`replicas_status`	Determines if data replicas across your cluster are lagging or synchronized correctly.
04	`select_query`	Executes read-only queries to pull analytics and insights from the database without making any changes.

See It in Action

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

U What were the top 5 product categories by revenue last quarter?

Q3 Revenue Breakdown (Millions USD)

CATEGORY	TOTAL REVENUE
Electronics	$42.1M
Software	$38.9M
Apparel	$15.6M
Services	$10.2M
Accessories	$7.8M

Data pulled from the sales_metrics table.

U Is the ClickHouse server currently online and ready for heavy querying?

Server Status Check:
✅ **Status:** ONLINE
🌐 **Latency:** 5ms
🕒 **Last Ping:** Just now.

The database is healthy and accepting queries. You can proceed with your analysis.

U I need to create a new table called 'user_activity' that tracks logins, user IDs, and timestamps.

Table Creation Result:
The `execute_query` action was successful.

- **Database:** default
- **Schema:** Created
- **Table:** user_activity
- **Columns:** (user_id: String, timestamp: DateTime)

The table is ready for data insertion.

Frequently Asked Questions

01 **How does the ClickHouse MCP help me run reports on my big data?**

It lets you ask natural language questions and instantly get analytical answers from massive datasets. You don't write SQL; your agent handles all the complex querying for you.

02 **Can I use the ClickHouse MCP to fix my database structure if something is wrong?**

Yes, if you need to add a column or rename a table, the MCP uses schema management tools so you can issue structural changes via conversation instead of logging into an IDE.

03 **What should I check first when starting analysis with ClickHouse MCP?**

Before pulling reports, always use the replication status tool. This confirms that all parts of your cluster are synced and ready to give you accurate results.

04 **Is this MCP just for reading data or can it write new information too?**

It does both. You can run read-only reports using select queries, but you can also use the execute query tool to make structural changes like creating new tables.

05 **If I'm a data engineer, how will ClickHouse MCP help me monitor my system?**

You gain real-time oversight of your entire cluster. You can instantly ping the server and check replication lag status without manual dashboard refreshing.

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": { "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 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 ClickHouse. 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	ClickHouse MCP
Server ID	`019eb8ad-7e75-7363-b4a6-5be2583160cf`
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/clickhouse.