MCP SERVER | NO CODE | CLOUD HOSTED

DataStax Astra DB Vector MCP for AI Agents

Run semantic searches and manage unstructured data collections.

DataStax Astra DB Vector gives your AI client direct conversational access to complex NoSQL databases and vector embeddings. It lets you perform everything from counting records to running semantic searches on unstructured data, all without writing code.

A+ Quality Score 100/100

nosql | vector-search | similarity-search | cassandra | unstructured-data | genai-infrastructure

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/datastax-astra-db-vector

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.

DataStax Astra DB Vector MCP

7 tools available
Cloud-hosted on Vinkius

Think of this MCP as a direct line into your database's guts. Instead of pulling up a console or writing multi-line queries, your AI agent talks to Astra DB naturally. You can ask it to count documents in an entire collection or find specific records using simple language.

Need to understand what's lurking in your unstructured data? Your agent runs vector similarity searches, finding documents that mean the same thing as a prompt, even if they don't share keywords. It also lets you manage the structure itself—you can list available collections and insert brand new JSON records with pre-generated embeddings.

This kind of deep, contextual access is huge for developers and data teams alike. When you connect this to Vinkius, your AI client gets a single point of entry to power all those complex operations. You're not just querying; you're managing the entire data lifecycle right from your chat window.

Core Capabilities

01 — List available collections

You can ask the MCP to list every collection currently active in your configured database namespace.

02 — Perform vector similarity searches

The agent runs Approximate Nearest Neighbor (ANN) searches, letting you find documents based on meaning rather than just matching keywords.

03 — Retrieve specific JSON records

You can ask the MCP to pull back one or multiple standard NoSQL JSON documents from any active collection.

04 — Insert new structured data

The agent creates and inserts a brand-new document, including pre-generated vector keys for embedding searches.

05 — Delete existing records

You can instruct the MCP to safely remove specific documents from a collection when they are no longer needed.

06 — Count total documents

The agent provides an accurate count of all active JSON documents across a specified Astra DB collection.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/datastax-astra-db-vector** — connect your AI agent in three steps.

01 Subscribe to the MCP and provide your specific Astra DB API Endpoint, Namespace, and Application Token.

02 Your AI client authenticates with the Vinkius platform and connects all those credentials securely.

03 You start asking natural language questions—like 'Find me documents about Q3 sales' or 'List my collections.' The agent translates that query into database actions.

The bottom line is, you talk to your database using the same conversational flow as a teammate over Slack.

Built For

DataStax Astra DB Vector serves developers and analysts who spend too much time context-switching between query builders, dashboards, and vector search tools. If you've ever needed to inspect data anomalies or run complex searches without writing a single line of boilerplate code, this MCP is for you.

Data Engineer

You use the agent to quickly debug JSON document anomalies and verify structural integrity by calling `list_collections` or running `count_documents` on demand.

AI Developer

You rely on vector similarity searches to retrieve precise, contextual embeddings for advanced RAG workflows without leaving your IDE environment.

Product Manager

You inspect unstructured vector data dynamically by performing targeted `vector_search` queries to understand how the AI interprets and surfaces search results behavior.

DBA

You manage records, knowing you can use conversational commands to find (`find_one_document`), retrieve (`find_documents`), or delete documents across collections effortlessly.

What Changes When You Connect

01 Contextual Data Access: You use `vector_search` to find documents based on meaning, not just keywords. This is a massive jump over traditional keyword filtering for your agent.

02 Full Lifecycle Management: The MCP lets you handle the entire document life cycle—you can `insert_document`, then later `delete_document` when data expires.

03 Structural Visibility: Need to know what collections exist? Use `list_collections` to map out your schema instantly, all through conversation.

04 Efficiency in Retrieval: Instead of running separate queries for counts and listings, you use `count_documents` or `find_one_document` directly via a single prompt.

05 Simplified Development: Developers can test complex data operations like `insert_document` right inside their chat window without writing boilerplate API calls.

Real-World Applications

Debugging RAG pipelines with vector search

An AI developer needs to know why a document is being missed during retrieval. Instead of manually running filters, they ask the agent to run a `vector_search` on the target collection, instantly surfacing nearby embeddings for debugging.

Auditing and cleaning up old data

A DBA needs to prune records that haven't been accessed in months. They use `list_collections` to identify the correct area, then instruct the agent to run a targeted `delete_document`, ensuring cleanup is accurate.

Prototyping new data ingestion workflows

A Product Team wants to test if new user feedback documents fit into the product catalog. They use `insert_document` with mock vector keys, validating the process before any real data hits the system.

Getting a quick inventory count of records

A team member needs to know if their latest batch upload succeeded. Instead of checking multiple dashboards, they simply ask the agent to run `count_documents` on the target collection for an immediate total.

Patterns to Avoid

Assuming simple text search is enough

✗ AVOID

Trying to find 'user dissatisfaction' by searching for only the word 'dissatisfaction'. The results are too narrow and miss related concepts.

✓ INSTEAD

You must use the `vector_search` tool. This performs semantic similarity, finding all documents that relate conceptually to 'user dissatisfaction,' giving you a much richer dataset.

Manually listing schemas for every query

✗ AVOID

Having to run separate commands just to find out what collections exist before starting work. This wastes time and slows down the process.

✓ INSTEAD

Start by asking the agent to `list_collections`. This gives you a complete map of your available data containers, letting you proceed immediately.

Trying to modify data without validation

✗ AVOID

Directly issuing commands to delete records across multiple collections without knowing which ones are active or if the record is critical.

✓ INSTEAD

Always check first by using `list_collections` and then use `find_documents` on a small sample set before you attempt any destructive action like `delete_document`.

The Right Fit

Use this MCP if your data needs are built around unstructured text, embeddings, or complex JSON documents. If you need to perform semantic searches (finding meaning), managing collections, or programmatically inserting/deleting records, this is the right choice.

Don't use it if all your data lives in highly structured relational tables and requires standard SQL joins for every query. For those cases, a dedicated relational database MCP will serve you better. Also, if you only need to read one specific document and never plan on querying or managing others, simpler document retrieval tools might suffice. But when the job involves both unstructured search *and* data maintenance, this is your best bet.

DataStax Astra DB Vector MCP: Semantic Search for Unstructured Data

Currently, running deep analysis on a database means jumping between multiple systems. You run a count query in one place, use a keyword filter in another, and then manually export data to a third tool just to perform vector lookups. It's slow, it involves tons of copy-pasting, and you always risk missing crucial context.

With this MCP, the whole process happens conversationally. You ask your agent to find documents related to 'customer retention strategies.' The system executes a `vector_search`, giving you immediate, semantically accurate results right in your chat. It’s pure, conversational insight.

DataStax Astra DB Vector MCP: Managing NoSQL Document Collections

If your data is decentralized—meaning you have records spread across dozens of different collections—you spend time just mapping the schema. You're always running `list_collections` to remember where that specific type of document lives, or trying to figure out if a record was created correctly using `insert_document`.

Now, your agent handles the overhead. It knows what collections are available and can help you manage them. Need to audit data? You tell it to count documents; need to test new inputs? Use the MCP to insert them for review. Everything is centralized.

7 Tools for DataStax Astra DB Vector: Document Operations & Embeddings

These tools let your AI client list collections, count records, perform semantic vector searches, or insert and delete specific documents in a NoSQL environment.

#	TOOL	DESCRIPTION
01	`list_collections`	Lists all available data containers (collections) within the connected Astra DB namespace.
02	`find_documents`	Retrieves multiple standard NoSQL JSON documents from a specified collection using filters.
03	`find_one_document`	Finds and returns a single, specific document within an Astra DB collection.
04	`vector_search`	Performs an Approximate Nearest Neighbor (ANN) search to find semantically related documents based on vector similarity.
05	`insert_document`	Creates and adds a new document into a collection, optionally including pre-generated vector data for embeddings.
06	`delete_document`	Removes targeted documents from an Astra DB collection after confirmation.
07	`count_documents`	Counts the total number of active JSON records present in a given collection.

See It in Action

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

U List all my current collections, then find me three user profiles from the 'users' collection.

✅ **Collections Found:**

- `user_vectors` (Active)
- `product_data` (Active)
- `system_logs` (Archived)

Here are 3 sample records from `user_vectors`:

```json
{
"user_id": "abc-123",
"city": "San Francisco",
"last_login": "2024-06-27"
}
```

Tip: If you want to search these profiles by what they mean, let me know!

U I need a document count for the 'product_data' collection and I also want to delete any record with an ID starting with 'test-'.

📦 **Document Count:**

Your `product_data` collection currently holds exactly 1,024 active JSON documents across the cluster.

🗑️ **Deletion Status:**

Successfully deleted 7 records matching the criteria (ID starts with 'test-'). The operation is complete.

U Find me some product descriptions that are conceptually similar to 'best practices for remote work setup'.

✨ **Vector Search Results:**
I found 5 highly related documents in the `product_data` collection. These documents scored highly on semantic similarity:

1. **Title:** Ergonomic Office Guide | **Score:** 0.92
2. **Snippet:** ...optimizing home office setups...
3. **Source:** Product Catalog v4

These results show the most contextually relevant information.

Frequently Asked Questions

01 **How can I use DataStax Astra DB Vector MCP to search documents by meaning, not just keywords?**

You simply ask your agent to run a vector similarity search. Instead of matching 'car,' it finds documents related to 'automobiles' or 'vehicle.' This gives you much deeper, contextual results from your unstructured data.

02 **Is DataStax Astra DB Vector MCP good for managing my database structure?**

Yes. You can use the agent to list all existing collections and count records across them. It lets you manage the overall shape of your NoSQL data without needing manual console access.

03 **Do I need a developer background to use DataStax Astra DB Vector MCP?**

No. You don't write code. You just talk to the agent using natural language, telling it what records you want to find or what data you want to add.

04 **Can I test new documents in DataStax Astra DB Vector MCP before going live?**

Absolutely. The agent allows you to insert and manage mock documents using the `insert_document` tool, letting you validate your data pipelines without touching production records.

05 **What if I want to find a single, very specific record?**

You can ask the agent to run a precise retrieval command (`find_one_document`). This is faster and more directed than searching through an entire collection of documents.

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 `"datastax-astra-db-vector": { "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

DataStax Astra DB Vector 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 DataStax Astra DB Vector. 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	DataStax Astra DB Vector MCP
Server ID	`019d7553-eb3a-736b-9627-acf7d69ef862`
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/datastax-astra-db-vector.