MCP SERVER | NO CODE | CLOUD HOSTED

Couchbase (Vector & NoSQL) MCP for AI Agents

Query and search complex JSON data structures in Couchbase

Couchbase (Vector & NoSQL) provides natural language access to complex, structured data in your Couchbase cluster. It lets you execute sophisticated N1QL queries against JSON documents and perform high-speed KNN vector similarity searches across massive datasets using only conversation.

A+ Quality Score 100/100

nosql | vector-search | knn | n1ql | embeddings | high-performance-computing

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

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.

Couchbase (Vector & NoSQL) MCP

7 tools available
Cloud-hosted on Vinkius

Need to talk to a database that handles everything from standard records to advanced semantic vectors? This MCP connects your entire Couchbase (Capella or self-hosted) cluster to your AI agent, giving you full control over both NoSQL data and complex vector storage through simple dialogue. Instead of writing boilerplate code for every search type, you just ask what you need.

This single connection lets your agent read document metadata using unique keys, run explicit SQL queries across entire collections, or find things based on meaning alone by mapping embeddings to vectors. If managing diverse data types—like structured records mixed with unstructured text and semantic similarity indexes—is a headache, this is what you need. You connect it via Vinkius, and suddenly your AI client can query the whole catalog of data sources using natural language.

Your agent won't just search; it will structure the results, pulling out exactly the JSON payload or specific field values you asked for.

Core Capabilities

01 — Querying structured NoSQL documents

You can run complex N1QL queries to pull specific JSON fields and records across your entire database structure.

02 — Finding similar items via vectors

Execute high-speed KNN vector searches, locating textual or semantic matches by mapping embeddings against existing indices.

03 — Retrieving specific document details

Fetch the full internal properties and data maps for any given document key within a collection.

04 — Searching full-text content

Perform text searches across large content trees using advanced Full-Text Search indexes.

05 — Auditing and navigating the database structure

Identify all existing buckets, scopes, and collections to understand how your data is organized in the cluster.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Couchbase URL, database username, and password.

02 Your AI client uses these credentials to connect directly to the specified Couchbase cluster endpoints.

03 You interact by asking natural language questions; the agent translates that into specific data operations like running N1QL queries or executing vector searches.

The bottom line is, you get instant access to query and search your entire NoSQL and vector dataset without writing database-specific code.

Built For

This MCP is for technical roles that deal with complex data models. Data Architects need it to verify schema consistency, Developers use it to prototype RAG features, and DBAs rely on it to audit search indexes across different environments.

Data Architect

They use this MCP to map out the entire data organization, listing buckets and scopes to verify that all collection boundaries are correctly set up before a major migration.

AI Developer

They test semantic matching for RAG applications by running vector searches on document embeddings and iterating through different index definitions.

Database Administrator (DBA)

They execute N1QL queries to check data consistency across multiple JSON structures and audit registered search indexes for performance bottlenecks.

What Changes When You Connect

01 Find exact documents quickly. By using the `get_document` tool, your agent retrieves complete internal properties for any unique key, eliminating manual lookups.

02 Move beyond simple keyword searches. The `vector_search` capability finds results based on meaning (similarity) rather than just matching words.

03 Handle complex data types easily. Use `execute_n1ql_query` to write sophisticated SQL-like commands that pull structured JSON payloads from deeply nested records.

04 Audit your setup thoroughly. Running `list_scopes` and `list_buckets` lets you map the entire database structure, knowing exactly where every piece of data lives.

05 Search everything text-based. The `fts_search` tool performs structural content matching across large documents, ideal for knowledge bases.

Real-World Applications

Debugging an application's search flow

A developer needs to check if a new feature correctly pulls related data. They ask their agent to `get_document` using the main user ID, and then use `list_indexes` to verify that all necessary lookup indexes are active before proceeding.

Finding policy documents by topic

Instead of keyword matching, a compliance officer needs to find policies related to 'GDPR data retention.' They ask the agent to perform a `vector_search` using an embedding for that topic, quickly returning semantically relevant files.

Extracting metrics from sales records

A product manager needs a report showing total revenue and item count. They prompt the agent with an N1QL query via `execute_n1ql_query` targeting specific fields, generating a precise JSON structure for analysis.

Mapping out data segmentation

A Data Architect needs to understand the boundaries of their system. They prompt the agent to run `list_buckets`, followed by `list_scopes` and `list_indexes`, generating a complete map of all available storage containers.

Patterns to Avoid

Treating data like simple key-value pairs

✗ AVOID

Trying to find related records by just asking for the document ID. This only returns basic metadata and fails when you need nested fields or aggregated results.

✓ INSTEAD

To pull complex, structured information, use `execute_n1ql_query` with N1QL constraints. This allows your agent to write specific queries that target deeply nested JSON structures.

Searching only by keywords

✗ AVOID

Asking the system to find 'sustainable gardening practices' using basic search terms. The results might be too narrow, missing conceptually related content.

✓ INSTEAD

Use `vector_search` and provide an embedding for the concept. This locates documents that are semantically similar (meaning-wise) even if they don't contain the exact words you typed.

Assuming all data is in one place

✗ AVOID

Sending a query without first verifying which collection holds the required records. This leads to vague errors or incomplete results.

✓ INSTEAD

Always start by running `list_buckets` and then `list_scopes`. Use this information to scope your subsequent queries, ensuring your agent targets the correct data boundaries.

The Right Fit

Use this MCP when your data is highly varied: you have both structured records (NoSQL), free-text content, and semantic vectors. If your primary need is just simple document retrieval by a known ID, a basic key-value store might suffice. But if you need to query *across* different data types—for example, finding documents based on a topic (vector search) and then pulling the resulting revenue figures (N1QL)—then this MCP is essential. Don't use it if your entire dataset lives in one simple relational database; those systems have dedicated tools for that purpose.

Couchbase (Vector & NoSQL): Managing Complex JSON Data with N1QL Queries

Manually querying a massive, hybrid data store involves jumping between multiple interfaces. You might use one tool to find the document ID, then another interface to pull related metadata, and yet a third place for complex reporting that requires filtering across dozens of fields. It's a constant cycle of context switching and copy-pasting.

With this MCP, your agent handles all that complexity internally. You simply ask it to 'Show me the names and prices of all travel packages under $500.' The system uses N1QL constraints, pulling the exact JSON payload you need without you ever seeing a single query or having to click through multiple tabs.

Couchbase (Vector & NoSQL): Semantic Search and Data Discovery

The old way of finding information was limited to keyword matching. If you searched for 'car battery replacement,' but the document used the term 'lead-acid power cell,' your search would fail, leaving you with a dead end.

Now, you ask the agent what it means to 'power an electric vehicle.' The system executes a vector search, finds documents that are conceptually related to EV power sources—even if they never mention the specific terms—and presents them directly.

Couchbase (Vector & NoSQL): 7 Tools for Data Querying

These tools allow your agent to interact directly with all components of the Couchbase cluster, from querying specific fields to finding vectors by similarity.

#	TOOL	DESCRIPTION
01	`list_buckets`	This tool identifies the major routing spaces within your database environment.
02	`list_scopes`	It retrieves detailed logging showing all defined scopes and collections available in your data structure.
03	`list_indexes`	This tool enumerates all active, structured search indexes attached to the cluster.
04	`execute_n1ql_query`	You can run complex queries using N1QL constraints to generate and retrieve specific JSON payloads.
05	`vector_search`	This executes a structured search, mapping structural KNN vectors to find semantic similarities via an index.
06	`get_document`	Fetch and retrieve the full internal mapped properties from specific Couchbase key-value documents.
07	`fts_search`	Perform structural text extractions by matching query strings against advanced content indexes.

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 main product types and their pricing?

Product Overview: Travel Samples

- **Economy Flight:** $85.00 (Min 1 person)
- **Local Tour:** $45.00 (Per adult)
- **Premium Package:** $320.00 (Includes transport & dinner)

U I need to find any documents related to 'cybersecurity compliance'.

Vector Search Results (Top 3 Matches)

RANK	DOCUMENT TITLE	SIMILARITY SCORE
1	Q3 Security Audit Report	0.98
2	Data Retention Policy v4.1	0.91
3	Compliance Checklist for APIs	0.75

The top result suggests a deep dive into security audits is warranted.

U List all available indexes so I know what's indexed.

I found three active search indexes for you:

- `customer-lookup`: Optimized for general user searches.
- `vector-index-v1`: Used for semantic similarity matching (KNN).
- `fts-content-search`: Handles full text extraction across all document bodies.

Which kind of search are you running next?

Frequently Asked Questions

01 **How do I query complex data structures using the Couchbase (Vector & NoSQL) MCP?**

You simply ask your agent to run a structured query. Instead of writing boilerplate SQL, you tell it what fields and criteria you need, and it executes N1QL constraints to deliver the exact JSON payload.

02 **Can this MCP find things that are related by meaning, not just words?**

Yes. You can use vector search capabilities within the Couchbase (Vector & NoSQL) MCP. This finds semantic matches using your embeddings, which is critical for advanced knowledge retrieval.

03 **Is this better than querying a traditional relational database?**

It handles variety better. If your data includes documents, JSON records, and vector metadata all in one place, the Couchbase (Vector & NoSQL) MCP manages that complexity for you, letting you treat it like one single source.

04 **What if I only have a document key? How do I get its data?**

You can use the dedicated retrieval tool to fetch the full internal properties of any specific document. This gives you all the associated metadata and content mapped to that unique key.

05 **How does Couchbase (Vector & NoSQL) help with data organization?**

The MCP lets your agent map out your entire cluster, listing buckets, scopes, and indexes. This gives you a clear audit of where all the different types of data are stored.

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 `"couchbase-vector-nosql": { "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

Couchbase (Vector & NoSQL) 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 Couchbase (Vector & NoSQL). 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	Couchbase (Vector & NoSQL) MCP
Server ID	`019d757d-2c1e-73bc-a0b5-f2e357f054af`
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/couchbase-vector-nosql.