MCP SERVER | NO CODE | CLOUD HOSTED

Marqo AI MCP

Control your entire semantic knowledge graph via chat.

Marqo AI (Vector Search & Embeddings) lets you manage entire semantic search infrastructures through natural conversation. You can run dense similarity searches, upload and index new JSON documents instantly, or audit your vector indices without writing complex API calls. Gain full control over document lifecycle management—from creating bounded indexes to deleting specific vectors.

A+ Quality Score 100/100

semantic-search | vector-embeddings | tensor-search | indexing | information-retrieval

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

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Marqo AI (Vector Search & Embeddings) MCP

6 tools available
Cloud-hosted on Vinkius

Connecting Marqo AI to your agent lets you manage semantic search infrastructure entirely via chat. You don't need to write boilerplate code just to check what data exists or how relevant a concept is. Instead, you simply ask your agent questions like, 'Show me all the indexes we have,' or 'Find the best product match for lightweight running shoes.' This MCP handles everything: it executes complex tensor searches against your stored knowledge, writes fresh JSON records into your indices instantly, and helps you manage the whole index lifecycle by creating new search boundaries. When you're ready to scale this capability across multiple platforms, remember that Vinkius hosts this MCP, giving your agent access to thousands of tools in one place.

Core Capabilities

01 — Perform semantic searches

Run natural language queries against your entire knowledge base to find highly relevant documents.

02 — Add new indexed data

Write fresh JSON records directly into your vector indices, making brand-new information immediately searchable by the agent.

03 — Manage index boundaries

Create explicitly defined vector indexes with custom rules and model settings for specific project needs.

04 — Audit index configurations

Retrieve detailed statistics, including document counts and embedding models, to check the health of your indices.

05 — Clean up old vectors

Delete specific documents or vectorized representations by targeting their unique IDs.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your Marqo API URL along with the necessary API Key.

02 Your agent connects these credentials, giving it immediate access to manage your vector search environment.

03 Start by asking your agent to list all available indexes or perform a semantic query from any MCP-compatible client.

The bottom line is you get full control over complex vector database operations using simple conversation prompts.

Built For

This connector is built for the data architect who gets frustrated having to manually write Python scripts just to check index stats. It's for ML Engineers needing real-time visibility into embedding results and developers who manage multiple, distinct knowledge bases.

Machine Learning Engineer

Uses this MCP to monitor vector index statistics and verify document embedding results directly from their workspace chat.

Data Architect

Manages the full lifecycle of multiple knowledge bases, using commands like create_index and list_indexes without touching a terminal.

Software Developer

Integrates AI-powered search into applications by adding documents with add_documents and maintaining data relevance with delete_documents.

What Changes When You Connect

01 Stop writing boilerplate code for basic checks. You can use list_indexes to see all available vector indices immediately, letting you know exactly what data sources your search needs.

02 The agent handles the complex math behind tensor_search. Instead of feeding it a query vector, you just ask a question in plain English and get highly relevant results back.

03 Keep your knowledge base clean using delete_documents. You target documents by ID to ensure that old or irrelevant vectors are physically removed from the index.

04 Need a dedicated search silo? Use create_index to build a new, bounded vector index with specific model rules, keeping unrelated data separate and optimized.

05 When you add_documents, your agent automatically handles embedding extraction. You just provide the JSON content; it becomes immediately searchable.

Real-World Applications

Updating a product catalog

A developer needs to update 50 new product descriptions in the vector store. Instead of writing a script, they simply ask their agent to use add_documents with the JSON data dump. The documents are indexed and available for search instantly.

Diagnosing a stale index

The ML Engineer suspects an old index is holding garbage data. They first call list_indexes, then get_index_stats on the target index to verify document counts before running delete_documents to clean out outdated records.

Building a feature store

A search architect wants a dedicated index just for user profiles. They use create_index first, setting up constraints, and then use add_documents repeatedly to populate it before testing with tensor_search.

Retrieving context from multiple sources

The agent needs to find the best shoe recommendation. It uses tensor_search on the 'products' index but first uses get_index_stats to confirm that index is running the correct embedding model.

Patterns to Avoid

Searching without knowing the source

✗ AVOID

The user tries to run a tensor search, but fails because they don't know if an index named 'user_data' or 'support_kb' exists in their Marqo setup.

✓ INSTEAD

Before searching, always start by running list_indexes. This shows you all available vector indexes so you can correctly target your query and avoid errors.

Trying to manage everything in one place

✗ AVOID

The user tries to update the index configuration and run a search using only general chat prompts, resulting in an ambiguous or failed operation.

✓ INSTEAD

Separate your tasks. Use get_index_stats first to audit what's there, then use create_index if you need a new boundary, and finally execute tensor_search.

Adding documents without structure

✗ AVOID

The user pastes raw text into the chat expecting it to be searchable, but the agent cannot properly vectorize or store it.

✓ INSTEAD

Always pass data in JSON format when using add_documents. This ensures all necessary fields for Marqo's vectorization process are present.

The Right Fit

Use this MCP if your core problem is semantic understanding and context retrieval from a large, structured knowledge base. If you need to answer questions like 'What was the main topic of Q3 reports?' or 'Find products similar to X,' this is your tool. Don't use it if you simply need to look up a single record by a known ID (that's better suited for a direct database connection). Also, don't rely on it just for simple text generation; its strength is in the retrieval step—it finds the context so your agent can answer intelligently. If you just want to run math or process structured data without searching, look into specialized tools that handle raw computation.

Managing knowledge bases used to feel like a series of disconnected API calls.

Today, setting up semantic search means juggling multiple interfaces. You have to write a script just to check if your index exists; another piece of code to read its stats before running the actual query. Then, if you want to add one new document, it's another function call with specific JSON headers and endpoints.

With this MCP, those manual steps disappear. Your agent handles the entire process in conversation. You just tell it what you need—whether that's finding a product using tensor_search or making sure your index is ready by calling list_indexes. It's seamless control over complex infrastructure.

Marqo AI (Vector Search & Embeddings) Gives You Complete Control.

You no longer need to manually run commands to check index configurations or list all collections. The agent handles the necessary calls, whether you use get_index_stats or simply ask, 'What indices do we have?'

The difference is that control feels intuitive. You're not managing code; you're directing a powerful search engine using natural language.

Marqo AI (Vector Search & Embeddings) with 6 Tools

These tools let you list indexes, run tensor searches, create new boundaries, add data, delete vectors, and check index statistics—all via your agent's chat interface.

#	TOOL	DESCRIPTION
01	`list_indexes`	Lists all the available vector indexes within your Marqo instance so you know what collections exist.
02	`get_index_stats`	Retrieves detailed configuration and operational statistics for a specific index.
03	`tensor_search`	Performs deep semantic similarity searches using natural language queries against your indices.
04	`add_documents`	Writes new structured documents into Marqo, making them instantly available for search.
05	`delete_documents`	Physically removes specific documents from the index by referencing their unique IDs.
06	`create_index`	Builds a new, isolated vector index with custom rules and constraints for a dedicated search purpose.

See It in Action

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

U Semantic search in index 'products' for 'lightweight running shoes for trails'

Executing tensor search... I've found 5 highly relevant products. Top match: 'Swift-Trail Runner' (Score: 0.89). Other results include specialized mountain gear and ultra-light hikers. Would you like to see the full JSON metadata for these matches?

U List all vector indexes in my Marqo instance

I've identified 3 indexes: 'products' (Ecommerce catalog), 'support-docs' (Technical KB), and 'user-profiles' (Personalization data). Which one would you like to check the stats for?

U Add this document to the 'support-docs' index: {"title": "API Auth", "content": "Use Marqo-API-Key header"}

Document added successfully to 'support-docs'. Marqo has vectorized the content using your configured embedding model. The document is now indexed and immediately available for semantic searches. Your new doc ID is 'marqo-12345'.

Frequently Asked Questions

01 **How does Marqo AI (Vector Search & Embeddings) MCP perform semantic searches?**

It uses the tensor_search tool to run dense similarity queries. You simply ask a question, and the agent handles turning that natural language into a vector query against your indexes.

02 **What should I use first when setting up Marqo AI (Vector Search & Embeddings)?**

Start by calling list_indexes. This action shows you every collection currently available on your instance, helping you map out your data landscape before running any queries.

03 **Can I update my knowledge base with Marqo AI (Vector Search & Embeddings) MCP?**

Yes, use the add_documents tool. You provide new JSON records to the agent, and it automatically processes them into your existing vector indices.

04 **Is there a way to isolate specific data sets in Marqo AI (Vector Search & Embeddings) MCP?**

You can use create_index. This tool builds an explicitly bounded, new vector index tailored for a very specific purpose or project.

05 **What if I find old documents that need removing in Marqo AI (Vector Search & Embeddings) MCP?**

Use delete_documents. You target the specific IDs of the vectors you want to remove, keeping your search index clean and highly relevant.

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 `"marqo-ai-vector-search-embeddings": { "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

Marqo AI (Vector Search & Embeddings) 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 Marqo AI (Vector Search & Embeddings). 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	Marqo AI (Vector Search & Embeddings) MCP
Server ID	`019d75cf-e8ce-737b-b6b7-cdb45ace1740`
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/marqo-ai-vector-search-embeddings.