MCP SERVER | NO CODE | CLOUD HOSTED

Zilliz Cloud MCP

Manage vectors and find what you need, naturally.

Zilliz Cloud MCP lets your agent manage and query vector data directly inside your AI workflow. You can list, create, drop, and describe entire collections in a cluster, all from natural language prompts. It executes high-performance vector similarity searches (ANN) while allowing you to filter results using complex metadata queries. Use it to power intelligent retrieval systems that understand context.

A+ Quality Score 100/100

similarity-search | vector-embeddings | ai-infrastructure | data-indexing | collection-management | milvus

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/zilliz-cloud

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.

Zilliz Cloud MCP

10 tools available
Cloud-hosted on Vinkius

This MCP connects your AI agent straight into your Zilliz Cloud vector database. Instead of writing boilerplate code or navigating complex dashboards, you talk directly to your data structure. Your agent handles all the heavy lifting—managing collections, inserting new records, and running deep similarity searches using just language.

For example, need to find documents related to 'Q3 financial compliance'? You tell your agent that, and it executes a precise search across your vector indexes. It even lets you refine those results by adding metadata filters, like only showing records from the 'Legal' department. If you're building complex data pipelines, connecting via Vinkius makes this MCP accessible to any compatible client, letting you treat your entire vector database management system as just another tool in your AI toolkit.

Core Capabilities

01 — Inventorying and Defining Collections

You can list all existing collections, describe a specific collection's schema, or create entirely new ones when setting up the infrastructure.

02 — Managing Data Lifecycle

Your agent can load entire collections into memory for faster searching, release them when done, and delete outdated records from your vector indexes.

03 — Ingesting and Modifying Data

The MCP allows you to insert new vector or scalar data into existing collections, or specifically delete entities that are no longer relevant.

04 — Performing Targeted Searches

You execute complex vector similarity searches using customizable metrics. You can combine this with metadata filtering to pinpoint exact records within massive datasets.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to the Zilliz Cloud MCP and provide your specific Zilliz Cluster Endpoint and API Key.

02 Next, invoke the MCP through your AI client. You tell your agent exactly what you need—for instance, 'List all collections' or 'Search for similar images related to X'.

03 The MCP executes the necessary operations against your cluster and returns a clean, structured result set directly into your chat window.

The bottom line is that it turns complex database operations into simple natural language commands executed by your agent.

Built For

Data Scientists and AI Engineers need this. If you regularly build Retrieval-Augmented Generation (RAG) systems, or if your application relies on understanding context from massive indexes, this MCP is essential. It lets you manage the data plumbing without writing a single line of API code.

AI Engineer

They test collection schemas and verify vector search parameters by simply asking their agent to run specific queries or describe a collection.

Data Scientist

They monitor cluster health, load collections for temporary analysis, and query data using metadata filters without writing boilerplate connection code.

Software Developer

They integrate vector database management directly into their development workflow by having their agent insert initial test data or drop old testing environments.

What Changes When You Connect

01 Stop writing repetitive boilerplate code. Instead of manually calling `list_collections` to check available indexes, just ask your agent—it handles the inventory step for you.

02 Drastically improve search accuracy by combining vector similarity searches with metadata filtering. Use a prompt to run `query_entities` and narrow results down immediately.

03 Manage data flow efficiently. If you only need temporary access to a dataset, use `load_collection` when starting the task and remember to call `release_collection` when finished.

04 Maintain clean indexes by having your agent execute cleanup tasks like using `delete_entities` or running `drop_collection` on outdated data sets.

05 Run complex retrieval operations simply. Instead of writing search parameters, describe what you need and let the agent run `search_vectors` to find similar items.

Real-World Applications

Identifying Data Sources for a New Feature

A developer needs to know what indexes exist before starting work. Instead of logging into the database console, they ask their agent to `list_collections`. The response instantly shows them all available vector collections, letting them start development immediately.

Finding Specific Documents in a Large Archive

A data scientist needs documents about 'Q2 marketing performance' but only from the 'West Coast' region. They ask their agent to `search_vectors` while simultaneously passing metadata filters, getting highly specific results without manual SQL joins.

Preparing for a New Data Model

An AI engineer wants to test a new data format. They use `create_collection` to build the schema and then run `insert_entities` with test vectors, all guided by conversational prompts.

Cleaning Up Old Development Environments

A team finishes testing an experimental dataset. Instead of manually deleting tables or running cleanup scripts, they prompt their agent to execute `drop_collection`, confirming that the old data is permanently gone.

Patterns to Avoid

Writing complex API calls for simple tasks

✗ AVOID

Manually structuring a JSON payload just to check if a collection exists, which requires understanding specific endpoints and HTTP methods.

✓ INSTEAD

Just ask your agent: 'List all vector collections.' The MCP handles the underlying `list_collections` call automatically.

Running searches without filtering

✗ AVOID

Executing a general `search_vectors` query that returns thousands of results, forcing you to sift through irrelevant data in your application.

✓ INSTEAD

Always refine the search using metadata filters. Prompt your agent to 'Search vectors for X only if the date is after Y' so it uses `query_entities` alongside the search.

Ignoring cleanup and resource management

✗ AVOID

Leaving temporary collections loaded in memory, which slows down other services or exhausts cluster resources over time.

✓ INSTEAD

After analysis is complete, prompt your agent to `release_collection` using the specific collection name. This frees up computational power.

The Right Fit

Use this MCP if your core problem involves semantic search, deep retrieval from unstructured data, or managing a vector index (i.e., you're building RAG systems). You need to ask: 'Is my primary goal finding the *meaning* behind the text?' If the answer is yes, this is your tool. Don't use this if you only need simple transactional CRUD operations on structured data (like updating a user's email address or tracking inventory counts). For those cases, stick with traditional relational database connectors; they handle strict schema enforcement better than vector indexes.

The headache of managing enterprise knowledge graphs

Today, figuring out what data is available involves a tedious cycle: logging into the dashboard, checking which collections exist, then writing specific API calls for each one just to confirm its schema. You spend time copying endpoints and wrestling with boilerplate code that simply tells you if your vector database is ready.

With this MCP, your agent takes over. You tell it what you want—like 'Show the schema for our customer reviews.' It executes the necessary `describe_collection` call internally, giving you a clean, readable answer instantly. The complex infrastructure work becomes a simple conversation.

Getting insights with Zilliz Cloud MCP

You no longer have to write separate scripts for different steps. Instead of writing one script to `list_collections`, another to `load_collection`, and a third to run the search, you tell your agent the full workflow in one prompt.

The difference is that your work moves from being defined by repetitive technical steps into a single, conversational objective. You get answers, not code snippets.

Zilliz Cloud: 10 Tools for Vector Search Management

These tools give your agent granular control over every aspect of your Zilliz vector database. You can manage collections, insert data points, or run complex similarity searches using a single chat interface.

#	TOOL	DESCRIPTION
01	`list_collections`	This tool retrieves a list of every collection currently stored in your Zilliz cluster.
02	`create_collection`	It creates a brand new vector collection within the cluster, requiring you to specify its structure.
03	`delete_entities`	This function removes specific data points or records from an existing collection.
04	`describe_collection`	It fetches detailed information, including the schema and status, for a single specified collection.
05	`drop_collection`	This tool permanently removes an entire vector collection from the cluster.
06	`insert_entities`	It adds new data, both vector and scalar, into a designated collection.
07	`load_collection`	This loads an entire specified collection into memory for immediate and optimized search availability.
08	`query_entities`	It searches for records using specific metadata filters (like dates or tags) to narrow down results.
09	`release_collection`	This function removes a collection from memory, freeing up cluster resources.
10	`search_vectors`	It performs the core task: executing a complex vector similarity search based on your configuration.

See It in Action

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

U List all vector collections in my Zilliz cluster.

I've retrieved your collections. You have 3 collections: 'image_embeddings', 'text_docs', and 'product_features'. Would you like more details on any of them?

U Show the schema and status for collection 'text_docs'.

Collection 'text_docs' is currently 'LOADED' with 150,000 rows. It has 3 fields: 'id' (Primary), 'vector' (FloatVector, 1536 dim), and 'metadata' (JSON).

U Drop the collection named 'old_data_backup'.

Successfully dropped the collection 'old_data_backup'. All associated data has been permanently removed from your cluster.

Frequently Asked Questions

01 **How do I start using Zilliz Cloud MCP to find my collections?**

You initiate the process by asking your agent to `list_collections`. This immediately returns a list of all available vector indexes in your cluster, letting you know where to focus.

02 **Can I search for data using Zilliz Cloud MCP with filters?**

Yes. You combine the powerful `search_vectors` tool with metadata filtering capabilities. This lets you narrow down billions of vectors to only those matching specific criteria, like date or department.

03 **What is the best way to update data using Zilliz Cloud MCP?**

You use `insert_entities` for adding new records. If a record needs correction or removal, you must explicitly tell your agent to run `delete_entities`.

04 **Does the Zilliz Cloud MCP handle data cleanup?**

Absolutely. You can use `drop_collection` to permanently eliminate old datasets, or use `release_collection` if you just need to temporarily free up memory resources without deleting the index.

05 **Is Zilliz Cloud MCP only for reading data?**

No. Beyond querying with `search_vectors`, this MCP allows full lifecycle management, including creating new indexes (`create_collection`) and inserting data (`insert_entities`).

06 **How do I find my Cluster Endpoint?**

You can find your Cluster Endpoint in the Zilliz Cloud Console under the 'Cluster Details' page. It typically looks like `https://in01-xxxxxxxxxxxx.vectordb.zillizcloud.com`.

07 **Why do I need to 'load' a collection before searching?**

Zilliz requires collections to be loaded into memory to perform high-performance similarity searches. Use the `load_collection` tool to make your data available for search.

08 **Can I filter my vector search using metadata?**

Yes, Zilliz supports hybrid search. You can use the `query_entities` tool for metadata-only filtering or include filtering expressions in your `search_vectors` JSON configuration.

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 `"zilliz-cloud": { "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

Zilliz Cloud 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 Zilliz Cloud. 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	Zilliz Cloud MCP
Server ID	`019d7628-8ef1-707e-b58c-635b098fbc22`
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/zilliz-cloud.