

MCP SERVER

NO CODE

CLOUD HOSTED

Qdrant MCP for AI Agents

Query vector embeddings and manage collections in your RAG pipeline.

Qdrant MCP lets your AI agent talk directly to your vector database. It handles similarity searches, collection management, and data inspection. Use it to debug your RAG pipeline and check your embeddings without leaving your chat window.

A+

Quality Score 100/100

vector-database

semantic-search

embedding-management

data-retrieval

vector-search



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

Qdrant MCP

7 tools available

Cloud-hosted on Vinkius

This Qdrant MCP connects your vector database to any AI agent and brings meaning-based search and database management into your conversation. Imagine you're building a RAG pipeline. Usually, you have to jump between your terminal, a dashboard, and your code to see if your embeddings actually look right or if your similarity search is hitting the right clusters. This MCP changes that. It puts your Qdrant instance directly into your AI's hands. Instead of writing manual scripts to check payload sizes or count points, you just ask your agent to do it. You can browse collections, pull specific points by ID, or even delete old data on the fly. It makes it way easier to debug your vector space during development. Because Vinkius hosts this, you can get it connected to your favorite client in minutes, letting you focus on the logic of your application rather than the plumbing of your database. You'll spend less time debugging connection strings or manually formatting JSON arrays for similarity queries. Your agent handles the technical heavy lifting, letting you see the actual results of your embedding logic immediately. Whether you're verifying a new index or cleaning up a production cluster, the interaction feels like talking to a teammate who already has the database credentials.

Core Capabilities

01 — Browse collections

List all your vector collections and see their distance metrics at a glance.

03 — Inspect point data

Read specific points by ID or scroll through large datasets to see their payloads.

02 — Run similarity searches

Perform nearest neighbor queries using a JSON array of floats to find relevant data.

04 — Clean up data

Delete redundant points from your collections without writing extra admin scripts.

05 — Check collection stats

Get total point counts and detailed metadata for any collection in your cluster.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to the Qdrant MCP and provide your Base URL and API Key.
- 02 Connect the MCP to your AI client like Claude, Cursor, or Windsurf.
- 03 Start asking your agent to query embeddings or manage collections directly.

The bottom line is you get direct, conversational access to your vector database's internal data and operations.

Built For

This is for the engineer who's tired of context-switching between a database console and their IDE. It's for anyone building RAG systems who needs to see what's actually happening inside their vector space.

AI & ML Engineer

Debugging RAG pipelines by inspecting embeddings and verifying similarity search results in real time.

Data Scientist

Testing distance parameters on live indices without needing to launch a Jupyter Notebook.

Backend Developer

Managing vector cluster configurations and clearing out bad datasets efficiently from the console.

What Changes When You Connect

- 01 Stop switching tabs by querying your vector database directly from your AI client.
- 02 Debug embedding quality faster by using scroll to inspect payloads in real time.

-
- | | |
|----|---|
| 03 | Manage your data lifecycle without writing extra admin scripts by using delete. |
| 04 | Verify your indexing numbers instantly with the count tool for any collection. |
| 05 | Get a bird's-eye view of your cluster's distance metrics with list_collections. |
-

Real-World Applications

RAG Pipeline Debugging

An engineer asks the agent to scroll through the 'docs' collection to see if the metadata is correctly attached to the vectors.

Data Cleanup

A developer tells the agent to delete all points in a 'test_batch' collection to clear space for a new run.

Cluster Audit

A data scientist asks the agent to list all collections to check which ones are using Cosine versus Dot product.

Point Verification

A dev asks the agent to get specific points by ID to verify that a specific user's profile was correctly embedded.

Patterns to Avoid

Manual Scripting

✗ AVOID

Writing a Python script every time you need to count points or list collections.

✓ INSTEAD

Use the count tool to get totals instantly or list_collections to see your cluster overview without writing any code.

Guessing Similarity

✗ AVOID

Not knowing if your search results are actually relevant to your query.

✓ INSTEAD

Use the search tool to test different float arrays and see the actual results immediately in your chat.

Data Bloat

✗ AVOID

Leaving old test data in your production index because you lack a quick way to remove it.

✓ INSTEAD

Use the delete tool to remove specific points from your collection as you go.

The Right Fit

Use this if you need a way to interact with Qdrant's data and operations through a natural language interface. It's perfect for RAG development and quick data auditing. Don't use it if you need to perform high-frequency, bulk batch inserts of millions of vectors. For that, stick to your native SDK or a dedicated ingestion pipeline. Use this when you want the agent to see and touch your vector data to help with development or troubleshooting.

Qdrant Vector Database Management

Most developers have to jump back and forth between their terminal, a dashboard, and their code to see if their vector embeddings are actually working. You spend half your time copy-pasting IDs or writing small scripts just to count how many points are in a collection.

This MCP lets your AI agent do that work for you. You can just ask it to show you the payload for a specific ID or list every collection in your cluster. It turns a multi-step manual audit into a single conversation.

Qdrant Semantic Search Testing

Testing similarity search parameters is usually a chore. You have to manually construct JSON arrays of floats, run them through a script, and then interpret the results to see if the distance metrics make sense.

Now you can just describe the search you want to your agent. It handles the vector input and shows you the matching results immediately. You get faster iteration on your RAG logic without the friction of manual testing.

Qdrant 7 Vector Database Tools for RAG

Query similarity, manage collections, and audit payloads with 7 specific Qdrant tools.

#	TOOL	DESCRIPTION
01	list_collections	Lists all collections in your Qdrant instance. It helps you see every vector set in your cluster at once.
02	get_collection	Retrieves detailed information about a specific collection. Use it to check distance metrics and point counts.
03	search	Performs a nearest neighbor vector search using a JSON array of floats. It lets your agent find the most relevant data points.
04	get_points	Retrieves specific points by their unique IDs. This is useful for checking if a particular record was saved correctly.
05	scroll	Returns points with their payloads for pagination. Use it to browse through large datasets without loading everything at once.
06	count	Counts the total number of points in a specific collection. It gives you a quick way to verify your indexing numbers.
07	delete	Deletes specific points from a collection permanently. Use it to remove old or incorrect data from your vector space.

See It in Action

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

U Show me the distance metrics for all my collections.



I've pulled the configuration for your Qdrant collections:

COLLECTION NAME	POINT COUNT	DISTANCE METRIC	VECTOR SIZE
docs-embeddings	4,500	Cosine	1536
image-features	120,400	Dot Product	512

Would you like me to inspect the payloads for `docs-embeddings` next?

U How many points are in the 'product_catalog'?



The `product_catalog` collection currently contains exactly **54,201** points.

U Scroll through the 'user_profiles' and show me the first 3 payloads.



Scrolling through `user_profiles`, here are the top 3 items:

1. ID: `1f2a3`

Payload: `{ "role": "admin", "department": "sales" }`

2. ID: `7b9c1`

Payload: `{ "role": "user", "department": "support" }`

3. ID: `9d8e4`

Payload: `{ "role": "user", "department": "sales" }`

Frequently Asked Questions

01 Can I use Qdrant MCP to manage my vector data?

Yes, you can list, count, and delete points directly. This allows you to manage your database content through a natural conversation with your AI agent.

02 How do I use Qdrant MCP for RAG debugging?

You can use it to scroll through payloads and check embedding quality. It helps you verify that your metadata is correctly attached to your vectors during development.

03 Does Qdrant MCP support similarity searches?

Yes, it can perform nearest neighbor searches using float arrays. This makes it easy to test how your agent retrieves information from your vector space.

04 Can Qdrant MCP help me clean up my vector database?

You can use the delete tool to remove specific points. This is great for clearing out test data or removing incorrect entries without manual scripts.

05 Is Qdrant MCP safe for production clusters?

It is a tool for interaction and auditing. While it allows for deletions, you should always be intentional with those commands in a production environment.

06 How do I connect Qdrant MCP to Claude?

You can connect it by adding your Qdrant Base URL and API Key in the Vinkius setup. Once connected, Claude can query your embeddings directly.

07 How do I find my Qdrant URL and API Key?

For **Qdrant Cloud**: Go to the Qdrant Cloud Console, select your cluster to open the Cluster Detail Page. The endpoint will be displayed there (e.g., `xyz.us-east4-0.gcp.cloud.qdrant.io`), and you can generate Database API Keys underneath it (they start with `eyJhb`). For **Self-hosted**: Provide your custom URL and the static custom key you defined in your `config.yaml`.

08 Can my AI use this for a RAG architecture directly?

Yes contextually, but practically your agent acts as the database debugger. It can formulate vector arrays to query `search_points`, retrieving identical payload structures. It's meant for the *engineer building the RAG*, helping you inspect distances and debug faulty retrieval mechanisms mid-code.

09 Does it support deleting vectors?

Yes. If an embedding got corrupted or references dropped articles, use the `delete` tool. Pass the collection name and the list of specific IDs. Qdrant handles the mutation instantly and updates the index without rebuilding.

10 What if I have millions of points?

Instead of overloading your chat context, instruct your agent to use the `count` tool to grasp the scale, and the `scroll` tool with a small limit constraint (e.g., 5-10 records at a time). This paginates large bodies cleanly when analyzing index health.

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 <code>"qdrant": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Qdrant 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 Qdrant. 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	July 2026
MCP Server	Qdrant MCP
Server ID	019d75fb-3c95-706b-a661-8961d9172ec3
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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/qdrant.