MCP SERVER | NO CODE | CLOUD HOSTED

Cohere MCP for AI Agents

Build Semantic Search and Conversational Chat with Vector Embeddings

Cohere connects enterprise-grade AI models directly into your workflow. Your agent can chat with advanced Command models for structured conversations, generate deep vector embeddings for semantic search, and re-rank large sets of documents to surface the most relevant information instantly.

A+ Quality Score 98.33/100

llm | embeddings | reranking | natural-language-processing | tokenization | chat-api

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/cohere

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.

Cohere MCP

6 tools available
Cloud-hosted on Vinkius

Building powerful applications that interact with complex text requires more than just a general language model. It needs specific tools for retrieval, understanding context, and structuring data. This MCP gives your AI agent direct access to Cohere's full suite of enterprise NLP capabilities.

Need to build a semantic search feature? Use the embeddings tool to turn documents into vectors, allowing your app to find meaning rather than just keywords. Want a conversational interface that cites its sources? Send messages via the chat API using Command models. If you're working with massive document sets and need to surface the absolute best result for a user query, you can re-rank them by relevance.

By connecting this MCP through Vinkius, your AI client treats Cohere like an internal utility—you don't switch between multiple API endpoints or write boilerplate HTTP code. You simply ask your agent to perform complex tasks, and it handles the full lifecycle: generating vectors, running a search, and presenting the final answer.

Core Capabilities

01 — Run structured conversations

Send multi-turn chats using Command models that provide text responses along with citations and tool call suggestions.

02 — Generate semantic vector embeddings

Create high-dimensional vectors for any text—be it a search query, document chunk, or classification label—for use in similarity search databases.

03 — Boost search relevance with reranking

Take a list of retrieved documents and apply advanced models to score them by how closely they match the user's original query.

04 — Inspect model capabilities

List all available Cohere models, checking their context lengths and specific use cases (like embedding or reranking).

05 — Measure text token usage

Estimate how many tokens a piece of text will consume before sending it to an AI model, helping manage costs and prevent overflow.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your Cohere API Key into Vinkius.

02 Connect your preferred AI client (like Cursor or Claude) to Vinkius, granting it access to the Cohere tools.

03 Ask your agent to perform a task—for example, 'Find documents about quantum computing and summarize them.' Your agent then automatically calls the necessary internal functions: listing models, generating embeddings, reranking results, and finally chatting with Command models for the summary.

The bottom line is that you get a single entry point into Cohere's entire suite of NLP tools, managed by your AI client.

Built For

This MCP is built for the ML Engineer needing to prototype advanced search pipelines or the Developer tasked with integrating robust document intelligence. If you're building anything that needs to understand context beyond simple keywords, this is your core utility.

ML Engineer

You use it daily to discover new models and generate embeddings with multiple types (float, int8, binary) while building out a vector database index.

Search Architect

Your job is optimizing search relevance. You rely on this MCP to re-rank documents after initial retrieval and manage text tokenization for accurate indexing counts.

Backend Developer

You need to quickly integrate enterprise-level chat functionality into an existing application without writing complex API orchestration code yourself.

What Changes When You Connect

01 Structured Conversations: Use the chat tool to interact with Command models, getting not just an answer but also source citations.

02 Advanced Retrieval: Generating embeddings via the embed tool lets you power true semantic search that goes far beyond basic keyword matching.

03 Search Precision: The rerank tool ensures that even if initial search results are broad, your users only see the most relevant documents first.

04 Efficiency Control: Before sending a query, use tokenize to check token counts. This prevents hitting API limits and saves credits.

05 System Visibility: List all available Cohere models using list_models so you always know which capabilities are on hand.

Real-World Applications

Building an Internal Knowledge Base Search

A developer needs to index thousands of internal PDFs. They use the embed tool to generate vectors for every document chunk, store them in a database, and then rely on the rerank tool when a user submits a query to surface the top three most relevant chunks.

Creating a Customer Support Chatbot

A support team wants an AI agent that answers complex questions using company manuals. They connect Cohere, use the chat tool with Command models for conversation, and utilize model discovery to ensure they are calling the right version of the chatbot.

Analyzing Large-Scale Research Papers

An ML researcher needs to compare concepts across 50 different papers. They use embeddings to generate vectors for key sections, allowing them to programmatically find conceptual similarities that manual reading would miss.

Optimizing Prompt Costs

A backend service needs to send many prompts but is worried about hitting token limits. It uses the tokenize tool first, checking the estimated length before making the actual API call and preventing costly failures.

Patterns to Avoid

Treating search as keyword matching

✗ AVOID

A user searches for 'best practices in cloud infrastructure' but only retrieves documents containing those exact three words, missing contextually similar content.

✓ INSTEAD

Don't rely on simple retrieval. Generate embeddings using the embed tool to create semantic vectors. This ensures your search engine finds conceptually related documents, not just keyword matches.

Ignoring model limitations

✗ AVOID

An engineer tries to send a 100k word document chunk for processing without verifying the model's context window.

✓ INSTEAD

Always use list_models first. This lets you confirm the maximum context length and ensure that your data chunks are sized correctly before calling any chat or embed tools.

Over-relying on initial results

✗ AVOID

A system displays 10 documents based on a raw vector similarity score without confirming relevance for the user's specific intent.

✓ INSTEAD

Always run rerank. After getting the top N candidates, use the rerank tool to apply an extra layer of scoring against the original query, guaranteeing the highest quality results are shown.

The Right Fit

Use this MCP if your application needs deep understanding of language: semantic search, conversational flow with citations, or complex document analysis. You need to measure concepts, not just words.

Don't use it if you only need basic text transformations (like simple character counts) or are building a system that doesn't involve large-scale document retrieval. For simple data storage and retrieval without deep semantic understanding, a standard key-value database might suffice. However, if your goal is to make the AI *think* about the meaning of the data, this MCP provides the necessary tools.

Cohere MCP for Advanced Document Search Relevance

Today, building a robust search engine means connecting disparate APIs: one service to chunk documents, another to generate vectors, and a third to run similarity queries. It's manual, brittle, and requires complex orchestration just to get a list of potentially relevant papers.

With this MCP, your AI agent handles the entire pipeline automatically. You simply ask it to find information about 'quantum computing,' and it manages generating embeddings, retrieving candidates, and using its advanced reranking models—delivering only the highest-quality results.

Cohere MCP for Conversational AI Context Management

Without this MCP, every interaction requires developers to manage state manually: passing conversation history back and forth in JSON payloads. This bloats the code, increases latency, and makes debugging a nightmare.

Now, your agent manages the complexity behind the scenes. It maintains context through chat commands, ensuring that follow-up questions are answered correctly because the tool handles the memory, letting you focus purely on the conversation's logic.

6 Cohere Tools for Advanced NLP and Vector Embeddings

Use these tools to control every step of the text processing workflow: from generating vectors to managing conversation state.

#	TOOL	DESCRIPTION
01	`chat`	Sends a message to a specified Cohere model and receives the text response, along with citations or tool call suggestions.
02	`detokenize`	Reconstructs readable text from an array of token IDs, which is useful for debugging encoding issues.
03	`embed`	Generates vector embeddings for various inputs, such as search documents or simple queries, suitable for similarity comparisons.
04	`list_models`	Retrieves a list of every Cohere model available, including their context length and specific capabilities.
05	`rerank`	Scores and reorders documents based on how relevant they are to a given query text.
06	`tokenize`	Breaks down raw text into individual tokens, allowing you to estimate the exact token count for API calls.

See It in Action

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

U I need to understand what kind of models Cohere has available for different tasks.

Here is a list of currently supported Cohere models. You can see which ones are best suited for chat, embedding creation, or ranking documents:

MODEL ID	PRIMARY FUNCTION	CONTEXT LENGTH	DETAILS
`command-r-plus`	Chat/Conversation	128K tokens	Best for complex reasoning.
`embed-v4`	Embeddings	N/A	Excellent for search vectors across multiple types.
`rerank-v3.5`	Reranking	N/A	Optimizes relevance scores on document sets.

U Hey, can you run the documents I gave you by 'machine learning models' and give me a score?

Top 3 Ranked Documents

1. **Transformers use attention mechanisms for sequence processing.** (Relevance Score: 0.95)
2. Neural networks are inspired by biological neurons. (Relevance Score: 0.72)
3. Python is a popular programming language. (Relevance Score: 0.12)

The analysis shows that the core concepts of attention mechanisms and neural network theory are most relevant to 'machine learning models.' The score difference indicates how far off the other documents fall.

U What's the best way to check if a document is too long for the chat model?

First, you need to pass the text through tokenization. This gives you an accurate count of tokens, which is much more reliable than just counting characters.

Token Analysis:

- Input Text: 'The rapid growth of AI requires better governance.'
- Tokens Generated: 12
- Detokenized Result: 'The rapid growth of AI requires better governance.'

Use this count against the model's context length to ensure you stay under budget and operational limits.

Frequently Asked Questions

01 **How does the Cohere MCP help me build a semantic search feature?**

The MCP allows your agent to generate vector embeddings for all your documents. Instead of matching keywords, the system finds meaning by comparing vectors, giving you deep contextual search results that feel natural.

02 **Do I need to write complex API calls every time my chatbot answers a question?**

No. Your agent handles all the complexity. You just chat with it naturally, and when it needs to fetch data or cite sources, the MCP automatically manages the internal tool calls.

03 **What is the difference between basic search and using Cohere's reranking?**

Basic search gives you a list of documents. Reranking takes that list and re-scores every document based on how well it actually answers the user query, putting the best result right at the top.

04 **Can I use this MCP to understand model limits or context sizes?**

Yes. By listing available models, you can check their specific capabilities and context lengths upfront. This prevents your application from failing due to hitting an invisible token limit.

05 **Is the Cohere MCP only for text? Can it handle other types of data?**

It focuses on advanced natural language processing tasks, dealing with documents and conversations. It uses vector embeddings to represent that meaning, which is key for sophisticated search.

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 `"cohere": { "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

Cohere 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 Cohere. 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	Cohere MCP
Server ID	`019d8427-e006-726d-9934-e74c17758f9a`
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/cohere.