MCP SERVER | NO CODE | CLOUD HOSTED

Cohere MCP for AI Agents

Building high-accuracy retrieval augmented generation (RAG) pipelines

Cohere gives your AI agents deep control over complex enterprise language processing. It lets you run the full lifecycle of generative AI tasks—from creating conversational responses to generating dense vector representations for semantic search, all through a single connection point.

A+ Quality Score 100/100

llm | generative-ai | natural-language-processing | chat-completion | reranking | ai-api

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

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 (AI Platform) MCP

5 tools available
Cloud-hosted on Vinkius

Building sophisticated AI workflows requires more than just a good chat model; it needs specialized tools for data handling and context management. This connector gives your agent complete control over the entire language pipeline. You can turn plain text into high-dimensional vectors, which powers advanced semantic search far beyond keyword matching. Need to improve Retrieval Augmented Generation (RAG)? Use this MCP to score and reorder documents based on how relevant they are to a given query, guaranteeing your agent pulls the best context every time.

It also handles foundational tasks like classifying incoming text into predefined categories or determining exactly what tokens are needed for specific models. Instead of jumping between multiple services, you manage everything—from initial data structuring to final model execution—all through natural conversation via Vinkius.

Core Capabilities

01 — Generate document vectors

Creates dense vector embeddings from any text input for semantic search and similarity matching.

02 — Prioritize research documents

Ranks multiple documents based on their semantic relevance to a specific user query, improving retrieval accuracy.

03 — Run conversational AI

Generates full conversational responses using advanced chat models for natural interaction.

04 — Analyze text structure

Breaks down text into precise token IDs that match a specific model's encoding dictionary.

05 — List available AI models

Retrieves a list of all current and available language models configured on your account plan.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and enter the necessary Cohere API Key (whether it's for testing or production).

02 Second, your AI client connects using that key. This grants immediate access to all text generation and language processing tools.

03 Third, you instruct your agent via natural conversation—for example, 'Generate embeddings for these three paragraphs,' or 'Rerank these search results.' The MCP executes the task immediately.

The bottom line is that once connected, Cohere provides a single point of access for managing complex, multi-stage generative AI workflows.

Built For

This connector targets technical builders who need to move beyond simple API calls. It's essential for the data scientist building robust RAG pipelines or the product team needing reliable text classification before a feature launch.

Data Scientist

Evaluates embedding quality and reranking performance in real-time to ensure knowledge bases provide accurate answers.

AI Developer

Tests and debugs complex text generation logic, ensuring conversational transformations work reliably across different model types.

Product Manager

Prototypes generative features using enterprise-grade language models to validate core product concepts before committing development resources.

What Changes When You Connect

01 Improve search results immediately. Use the `rerank_documents` tool to ensure your agent pulls the absolute most relevant document chunk, boosting RAG accuracy.

02 Stop relying on basic keyword matching. Generating embeddings with `generate_embeddings` turns text into vectors that capture true semantic meaning for deep searching.

03 Handle complex conversations easily. The `chat_completion` tool allows your agent to manage multi-turn dialogues and maintain conversational context naturally.

04 Verify model compatibility before deployment. Use `list_models` to check available hashes and identifiers, preventing runtime API failures.

05 Refine text inputs precisely. By using the `tokenize_text` tool, you can audit exactly how a specific model interprets your input data.

Real-World Applications

Building an internal knowledge search engine

A data scientist needs to build a system that searches company manuals. Instead of simple keywords, they use `generate_embeddings` on the manual chunks and then instruct their agent to `rerank_documents` against a user query, ensuring only the top three most relevant pages are returned.

Creating customer support chatbots

A product team is building a chatbot that needs to handle complex queries. They use `chat_completion` for basic Q&A and supplement it by first using `list_models` to select the best model for specific tasks, ensuring the conversation feels natural.

Automating content moderation

An engineer needs a system that can categorize user-submitted reports. They use the input classification capabilities (via chat completion) to automatically label text—'Spam,' 'Support,' or 'Sales'—before it hits the database.

Developing complex multi-step data pipelines

A developer needs to ensure a system can handle both conversational chat and structured token processing. They use `chat_completion` for dialogue, then follow up with `tokenize_text` to confirm the model's input limits.

Patterns to Avoid

Assuming simple search works

✗ AVOID

The agent simply queries a document chunk and trusts the first result it gets, even if the overall context is poor or irrelevant.

✓ INSTEAD

Always run `rerank_documents` on your search results. This process scores all retrieved documents against the query, guaranteeing you surface the most semantically accurate information.

Ignoring model limitations

✗ AVOID

Sending a massive block of text to the chat function without knowing the token limits for that specific model, causing unpredictable failures.

✓ INSTEAD

Before running anything, use `list_models` to confirm which models are available. Then, if needed, use `tokenize_text` on your input data to audit its exact token count.

Treating text as just words

✗ AVOID

Only using the raw text in a search query without converting it first. The system fails because it treats 'apple' and 'banana' as unrelated.

✓ INSTEAD

Always run `generate_embeddings` on your source text chunks. This process converts words into mathematical vectors, allowing the AI to understand relationships like similarity.

The Right Fit

Use this MCP if your workflow requires deep understanding of language—meaning you need to know *what* a piece of text means (embeddings), or you need to prioritize contextually rich information (reranking). You should use it when building complex RAG systems, advanced chatbots, or specialized data pipelines. Don't use it if all you need is simple API call logging or basic CRUD operations; those are handled by other connectors. If your only goal is text generation and you don't care about context retrieval, a general-purpose LLM tool might suffice, but for enterprise grade results, this MCP provides the necessary structure.

Cohere (AI Platform) MCP: Mastering Document Context Retrieval

Manually building an effective knowledge retrieval system is a nightmare. Today, developers often have to copy document chunks into separate services just to check for relevance before passing them to the main LLM. They spend hours tweaking scoring algorithms and testing different indexing methods.

With this MCP, you skip that manual orchestration. You simply instruct your agent to `rerank_documents`. It takes a list of potential source documents and instantly reorders them by semantic fit against the query. You get clean, prioritized context right in your workflow.

Cohere (AI Platform) MCP: Structuring High-Quality Text Outputs

If you're building a system that outputs structured data or needs to process text with strict rules, it used to require multiple validation layers and custom parsing logic. You had to write boilerplate code just to ensure the output was in the right format.

Now, your agent handles this natively. By integrating these capabilities through Vinkius, you can manage complex tasks like classifying input data or tokenizing text for model auditing—all within one conversation. The result is predictable, reliable AI outputs that actually integrate into production systems.

Cohere (AI Platform) MCP: 5 Tools for Text & Embedding Ops

These tools let your agent generate vector embeddings, reorder search results by relevance, run conversations, and manage model details.

#	TOOL	DESCRIPTION
01	generate_embeddings	Creates dense vector embeddings from text, allowing your agent to understand the meaning behind phrases.
02	rerank_documents	Compares a query against several documents and reorders them by semantic relevance for better context retrieval.
03	chat_completion	Runs formatted conversational transformations to generate natural, back-and-forth text responses using advanced chat models.
04	tokenize_text	Breaks down specific text into integer segments that match the active token dictionary of a chosen model.
05	list_models	Checks and lists all available language models and identifiers on your current plan to ensure API compatibility.

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 find the top three most relevant documents about quantum computing from this folder.

Top Results for Quantum Computing

1. **Qubit Stability in Cold Environments:** (Relevance: 98%) Discusses superconducting circuits and thermal management.
2. **Quantum Entanglement Metrics:** (Relevance: 95%) Details measurement techniques for entangled particle pairs.
3. **Future Applications of QC:** (Relevance: 88%) Outlines potential uses in drug discovery and materials science.

Confidence Score: High. The results are highly focused on the core physics principles.

U Can you generate embeddings for these five product features?

Embeddings Generated Successfully!

FEATURE	VECTOR ID (EXCERPT)
Real-time analytics dashboard	[0.12, -0.55, 0.91, ...]
Multi-region data synchronization	[-0.33, 0.87, -0.12, ...]
Customizable user roles	[0.60, 0.01, 0.45, ...]
HIPAA compliant storage	[-0.99, 0.22, -0.77, ...]

These vectors are ready to index for your semantic search feature.

U Write a short conversational guide on the benefits of using this platform.

Hey! Sure thing. This platform helps you take complex language tasks and turn them into simple, natural conversations.

Think of it like this:
• **Before:** You had to write complicated prompts for every single task (summarize, classify, search).
• **Now:** Your agent handles the complexity behind the scenes. It uses advanced models to chat with you while knowing exactly how to pull and prioritize the best source material.

It's all about making powerful AI feel easy.

Frequently Asked Questions

01 **How do I make my chatbot retrieve accurate context using Cohere (AI Platform) MCP?**

You improve accuracy by running the document results through a reranking process. Instead of relying on simple search, you let the tool score all retrieved documents against the query to guarantee only the most relevant information is passed to the chat model.

02 **Does Cohere (AI Platform) MCP help with semantic search?**

Yes, it does. You use the embeddings function to convert your text into numerical vectors. This allows your agent to understand that two phrases mean the same thing, even if they don't share keywords.

03 **What kind of tasks can I automate using Cohere (AI Platform) MCP?**

You can manage everything from basic conversational chat completions to complex data auditing. This includes classifying incoming text or listing available models for deployment checks.

04 **Is this connector suitable for enterprise RAG systems?**

Absolutely. It provides the core components needed for robust RAG, specifically document reranking and embedding generation, which are critical for reliable knowledge retrieval in corporate environments.

05 **Do I need to write custom code for every text processing step with Cohere (AI Platform) MCP?**

No. You manage the entire workflow—from generating embeddings to running chat completions—through natural conversation in your agent, eliminating much of the manual API orchestration.

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-ai-platform": { "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 (AI Platform) 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 (AI Platform). 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 (AI Platform) MCP
Server ID	`019d7577-24f0-71aa-b4c4-e41f73b6ef1c`
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-ai-platform.