MCP SERVER | NO CODE | CLOUD HOSTED

Lucidworks Fusion MCP

Control your enterprise knowledge graph via AI chat.

Lucidworks Fusion provides full control over your corporate knowledge graph through natural conversation with your AI agent. Use this MCP to execute complex semantic searches, monitor machine learning ranking jobs, and update document indexes for deep enterprise discovery.

A+ Quality Score 100/100

enterprise-search | semantic-search | machine-learning | vector-search | data-ingestion | search-ranking

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/lucidworks-fusion-ai-search-discovery

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.

Lucidworks Fusion (AI Search & Discovery) MCP

10 tools available
Cloud-hosted on Vinkius

This connector lets you take the complexity out of running an enterprise search platform. You can instruct your AI client to perform advanced queries that go beyond simple keyword matching—you'll run vector-based searches against specific documents or apps. Need to improve how relevant your search results are? Your agent handles sending user behavior signals, like clicks and conversions, directly into the system's machine learning models for automatic ranking improvements. Furthermore, you can keep your data fresh by syncing brand new document mappings or auditing existing records in your physical search collections. Because Vinkius hosts this MCP, you connect once to access powerful tools designed specifically for Search Engineers and Data Scientists who need granular control over their infrastructure.

Core Capabilities

01 — Run advanced semantic searches

Perform complex queries using both keywords and AI vectors against specific application profiles.

02 — Improve search relevance with user data

Send clickstreams or conversion signals to feed the system's machine learning models, making future searches better.

03 — Audit and manage document indexes

Update entire textual mappings or check which underlying search indices are active across your tenant.

04 — Monitor ML training jobs

Track the status of background data ingestion or machine learning model training to confirm everything is processing correctly.

05 — Inspect system configuration

List and audit how different query and index profiles are set up, allowing you to understand your search routing rules.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/lucidworks-fusion-ai-search-discovery** — connect your AI agent in three steps.

01 Subscribe to this MCP on Vinkius.

02 Provide your Lucidworks Host URL and API Token credentials.

03 Use natural language commands through your AI client to execute search, index management, or job monitoring tasks.

The bottom line is you get to control high-level enterprise data architecture using only conversation, without writing a single API call.

Built For

This MCP is for the Search Engineer who needs to audit query profiles and index status instantly. It's built for Data Scientists monitoring ML job health and Digital Experience teams who need deep visibility into document search results across multiple apps.

Search Engineer

Uses the MCP to test complex queries or verify indexing results against specific profiles without writing manual API scripts.

Data Scientist

Monitors machine learning job statuses and verifies signal ingestion to confirm that ranking models are trained on real user behavior data.

Digital Experience Manager

Audits search results across different applications to pinpoint why certain documents aren't surfacing correctly for users.

What Changes When You Connect

01 Stop logging into multiple dashboards to check search health. You can now list and audit underlying search indices and physical shards using a single command, giving you full visibility into your data distribution.

02 You don't need to manually write complex API payloads for testing. Use the MCP to execute deep, custom JSON logic that overrides standard Solr vectors natively, making query debugging instantaneous.

03 Improve relevance without manual model retraining. By using tools like `lw.post_signal`, your agent sends user behavior data (clicks, conversions) directly into Fusion's ML pipeline, improving search results automatically over time.

04 Audit the rules governing your search logic from one place. You can list and inspect query profiles to understand exactly how AI models are configured in your routing layers, saving hours of manual documentation review.

05 Track data integrity effortlessly. Monitor active ML training jobs or check index profiles directly through conversation, ensuring that critical background processes aren't failing silently.

Real-World Applications

Debugging poor search ranking for a new feature

A Digital Experience Manager notices search results are missing key documents. They ask their agent to audit the query and index profiles, running `lw.list_query_profiles` first, then using `lw.query_filtered` to structurally extract properties, immediately pinpointing which data fields aren't being indexed correctly.

Validating ML model performance after a traffic spike

A Data Scientist suspects the ranking model is outdated. They ask their agent to list active ML training jobs using `lw.list_jobs` and then send simulated user clicks via `lw.post_signal`. This confirms that the system is receiving fresh signals needed for accurate re-ranking.

Onboarding a new data source into search

A Search Engineer needs to add a whole batch of new documents to the index. Instead of writing a bulk API call, they instruct their agent to use `lw.index_documents`, confirming that the process runs and extracts rich churn flags from the newly uploaded records.

Troubleshooting data gaps in reporting

A team needs to see what collections are active for billing purposes. They ask their agent to execute `lw.list_collections`, which enumerates all attached structured rules, providing an immediate and clear view of the connected data sources.

Patterns to Avoid

Writing boilerplate API calls

✗ AVOID

Opening a terminal window and pasting complex CURL commands to check if indices are running or what profiles exist. It's tedious, error-prone, and takes minutes of setup time.

✓ INSTEAD

Just ask your agent: 'What query profiles are active?' The tool `lw.list_query_profiles` handles the complexity for you via natural conversation.

Checking data sources manually

✗ AVOID

Having to jump between documentation pages and separate admin consoles just to confirm if a specific billing collection is even attached or exporting correctly.

✓ INSTEAD

Use `lw.list_collections` with your agent. It gives you an immediate, aggregated list of all explicitly attached structured rules.

Assuming data freshness

✗ AVOID

Running a search query only to find out later that the underlying document index hasn't been updated in weeks, causing stale results.

✓ INSTEAD

Before running any deep query, ask your agent to run `lw.list_index_profiles` first. This verifies which arrays are spanning native hold parsing before you start.

The Right Fit

Use this MCP if your core problem is understanding the *architecture* and *health* of a complex enterprise search system. You need to audit query profiles, monitor ML jobs, or ingest data at the index level. If your job involves validating how deeply custom JSON logic maps over Solr vectors, this is your tool. However, don't use it if you just need simple document lookup; for basic retrieval, a standard vector search connector will suffice. You should also avoid using it if you only need to manage user accounts or billing information; those are separate identity management tools. This MCP lives squarely in the domain of Search Engineering and Data Science.

Checking your enterprise knowledge base used to feel like archaeology.

Today, understanding why a document isn't appearing when it should is a nightmare. You have to jump between the search admin console, check documentation pages for index schemas, and then manually run API calls just to list what profiles are even available. It’s slow, you spend half your day copy-pasting parameters, and by the time you finish, you still aren't 100% sure if the data is fresh.

With this MCP, that entire process collapses into a conversation. You tell your agent to audit the system, and it uses tools like `lw.list_query_profiles` or `lw.list_collections`. It synthesizes all that complex backend information—the rules, the profiles, the collections—and gives you an immediate answer.

Controlling Search Indexing with lw.index_documents

Before this MCP, updating a large set of documents meant coordinating massive API batches and hoping that the validation process extracted every necessary flag correctly. It was a high-risk operation requiring specialized scripting knowledge.

Now, you simply instruct your agent to run `lw.index_documents`. It handles the complex logic internally, running irreversible validations and extracting rich churn flags across your entire dataset. You get reliable data integrity without writing one line of code.

Lucidworks Fusion (AI Search & Discovery) with 10 Tools

Use these tools to control every aspect of your enterprise search platform, from querying specific document profiles to managing machine learning training jobs.

#	TOOL	DESCRIPTION
01	`lw.list_collections`	Lists the structured rules that are actively exporting billing data.
02	`lw.post_custom_query`	Allows you to inject highly customized JSON logic that overrides standard Solr vector processing.
03	`lw.query_filtered`	Performs a structural extraction of properties necessary for managing active accounts.
04	`lw.list_index_profiles`	Identifies and lists the precise active arrays that handle native hold parsing.
05	`lw.index_documents`	Performs an irreversible data validation process, extracting rich churn flags from documents.
06	`lw.list_jobs`	Identifies and lists the precise active arrays related to native gateway authentication.
07	`lw.list_query_profiles`	Runs an automated validation check that routes specific gateway history data.
08	`lw.query_search`	Resolves precise AI vector rules matching strict profiles, identifying records within the platform's core system.
09	`lw.post_signal`	Retrieves explicit cloud logging information while tracing vault limits.
10	`lw.query_sorted`	Generates a highly available JSON payload with hard customer bindings, sorted by date descending.

See It in Action

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

U Search the 'Support' app for 'password reset' using the 'default' profile

Executing search in Fusion... I've retrieved 5 relevant documents. Top result: 'How to reset your corporate password' (Score: 0.95). Other matches include troubleshooting steps and security policies. Would you like to see the full document content?

U List all active ML training jobs for the 'Commerce' application

I've identified 2 active jobs for 'Commerce': 1) 'Query-Intent-Model' (Status: Running, 45% complete), 2) 'Click-Re-Ranking' (Status: Waiting). I'll let you know once the intent model training is finished. Would you like to check the index profiles as well?

U Post a signal: user clicked on doc ID 'doc-987' in the 'Support' app

Signal posted successfully. I've sent the click event for 'doc-987' to the Fusion signal pipeline for the Support app. This will be used to improve future search results for similar queries.

Frequently Asked Questions

01 **How do I check which document collections are active using Lucidworks Fusion MCP?**

You use the `lw.list_collections` tool with your agent. This command enumerates all explicitly attached structured rules, giving you a comprehensive list of every active data source for billing purposes.

02 **Can I check my ML job status using Lucidworks Fusion MCP?**

Yes, use `lw.list_jobs`. This tool identifies and lists the precise active arrays spanning native Gateway authentication, letting you confirm if your machine learning models are training correctly.

03 **What is the best way to improve search results with Lucidworks Fusion MCP?**

You should use `lw.post_signal`. This sends explicit cloud logging data, allowing you to feed user actions like clicks directly into the system for continuous improvement of search relevance.

04 **Does the Lucidworks Fusion MCP let me test custom queries?**

Absolutely. The `lw.post_custom_query` tool lets you inject deeply customized JSON logic that overrides Solr vectors natively, allowing for highly specific testing of your search parameters.

05 **I need to see all available index profiles in the Lucidworks Fusion MCP.**

Run `lw.list_index_profiles`. This tool identifies and lists the precise active arrays spanning native Hold parsing, giving you a map of your current indexing structure.

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 `"lucidworks-fusion-ai-search-discovery": { "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

Lucidworks Fusion (AI Search & Discovery) 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 Lucidworks Fusion (AI Search & Discovery). 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	Lucidworks Fusion (AI Search & Discovery) MCP
Server ID	`019d75ca-c3c3-73cc-95c9-a0c8ad6c6f6a`
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/lucidworks-fusion-ai-search-discovery.