MCP SERVER | NO CODE | CLOUD HOSTED

Klevu (E-commerce AI Search) MCP

Master product discovery and merchandising logic.

Klevu AI Search MCP powers your e-commerce product discovery using natural conversation. Ask your agent to perform complex searches, audit category layouts, or fetch specific recommendations just by talking to it. You can execute keyword lookups, apply precise facet filters (like brand or size), and monitor global trending items without writing a single API call.

A+ Quality Score 100/100

site-search | product-discovery | merchandising | ai-search | recommendation-engine | search-relevancy

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/klevu-e-commerce-ai-search

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.

Klevu (E-commerce AI Search) MCP

10 tools available
Cloud-hosted on Vinkius

Connecting Klevu through this MCP lets you treat your e-commerce catalog like a conversation. Instead of building complex JSON payloads in Postman or digging through developer documentation, you talk to your agent about what you need—whether it's finding every jacket under $100, checking if a specific category page displays related items correctly, or seeing what products are currently spiking in popularity across the whole store.

It takes search logic and merchandising rules out of the code and into plain English. This level of control means you can quickly audit product rankings and recommendation setups for your site. By hosting this tool within the Vinkius catalog, you get immediate access to advanced discovery tools alongside any other e-commerce service your team uses.

Core Capabilities

01 — Conducting keyword searches

Find product listings across your entire catalog using natural language keywords.

02 — Filtering search results

Narrow down large result sets by applying specific characteristics like color, size, or brand.

03 — Checking category merchandising rules

Retrieve products configured for a specific category path to audit how your site displays content.

04 — Getting product recommendations

Fetch machine learning-driven suggestions, such as items frequently bought together or visually similar goods.

05 — Monitoring global trends

View the most relevant and fastest-selling products across your store to spot market opportunities.

06 — Running custom search payloads

Execute deeply nested, specific API queries using raw JSON structures.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/klevu-e-commerce-ai-search** — connect your AI agent in three steps.

01 Subscribe to this MCP and enter your Klevu Search URL and API Key.

02 Your AI agent uses the provided credentials to connect to your live e-commerce data.

03 You simply prompt your agent with a request, like 'Show me all blue shoes under size 10,' and it executes the search logic for you.

The bottom line is that this MCP turns complex product discovery into a simple conversation with your AI client.

Built For

This tool is critical for Digital Merchandisers and E-commerce Developers who spend too much time manually testing search paths or writing repetitive API calls. It's also perfect for Data Analysts who need to monitor product performance trends without logging into a separate dashboard.

Digital Merchandiser

Audits the display logic on specific category pages and verifies that recommended products show up correctly based on current merchandising rules.

E-commerce Developer

Tests search relevance, runs custom API queries, or validates product details for integration into a client application, all through conversational prompts.

Data Analyst

Monitors global trending products and analyzes search performance to identify gaps in the catalog or untapped market interest.

What Changes When You Connect

01 Bypass manual testing. Instead of repeatedly running separate queries for different filters, you can use the `search_filtered` tool to check complex combinations—like 'red shoes' *and* 'size 9'—all in one conversational prompt.

02 Optimize site performance by monitoring what sells best right now. Use the `search_trending` tool to identify global product velocity and quickly pinpoint seasonal opportunities, avoiding guesswork about inventory needs.

03 Control how products are displayed on your site. With `search_category`, you can audit whether a specific category page is correctly fetching all necessary related items according to your merchandising rules.

04 Go beyond basic searches. If the AI needs deep data—say, comparing multiple product attributes simultaneously—the `search_raw` tool lets you execute complex JSON payloads without needing specialized coding knowledge.

05 Improve user experience instantly. You can test how fast and accurate search results are by using `search_autocomplete`, ensuring that partial terms still guide users to the right products.

Real-World Applications

Checking for product gaps during a seasonal launch

A merchandiser needs to confirm if their new fall collection is showing up correctly across multiple related categories. They ask, 'Show me all items in the Winter Outerwear category that are blue.' The agent uses `search_category` and `search_filtered` together to provide a precise list of what's currently visible.

Debugging complex site search issues

A developer notices that 'waterproof jacket size 10' sometimes fails. Instead of writing multiple API calls, they ask the agent to run a targeted query using `search_filtered` and get a clean list, instantly confirming if the facet combination is supported.

Analyzing competitor product positioning

A data analyst wants to see what products are currently gaining traction globally. They ask the agent to run `search_trending`, getting an immediate overview of top sellers and high-demand items that they can use for inventory planning.

Building a recommendation engine prototype

A developer wants to test how product suggestions look. They ask the agent to execute `search_recs` on an existing search result, getting machine learning predictions without writing any backend code or managing external services.

Patterns to Avoid

Assuming a simple keyword search is enough

✗ AVOID

Thinking that just searching for 'running shoes' will give them all the necessary data, when in fact they need to narrow it down by size and color.

✓ INSTEAD

Don't stop at basic keywords. Use `search_filtered` immediately after running a general search to apply specific facets like brand or size. Then use `search_sorted` if you want the results ordered by price.

Manually building every API request

✗ AVOID

Spending hours in Postman, writing separate JSON payloads for category checks, recommendation fetching, and keyword searches.

✓ INSTEAD

Use your AI agent. Simply ask the question: 'Check the recommended products on this page.' The agent handles the complex underlying logic, including calling `search_recs` or `search_category`, giving you a clean answer.

Forgetting about pagination limits

✗ AVOID

Running a search and only seeing 20 results, then assuming those are the only ones available.

✓ INSTEAD

After any search query, always follow up by asking for paginated results. The agent uses `search_pagination` to ensure you see everything available in the catalog.

The Right Fit

Use this MCP if your core workflow involves discovery and auditing e-commerce logic. You need to know *why* a product appears where it does, or what happens when users search for complex combinations of attributes (e.g., 'leather boots' + 'size 12'). This is the right tool if you need conversational control over highly structured data sets.

Don't use this MCP if your only goal is to retrieve a single piece of static information, like fetching details for a known product ID—for that, `search_product_id` works perfectly. Also, don't use it if your task involves managing user accounts or processing payments; those require different types of APIs.

If you are struggling with complex filtering and need to check how

products behave across categories, this MCP is the definitive solution. Otherwise, stick to a basic API connector.

The Headache of Testing Search Paths

Right now, testing product discovery feels like running an elaborate series of manual clicks or spending hours in Postman. You have to build one query for keywords, another payload just to filter by color, and then a third one just to check the category merchandising rules. It’s repetitive, slow, and frankly, it makes you hate your job.

With this MCP, that entire process disappears. You tell your agent what you want—for example, 'Show me all running shoes available in red.' The system executes multiple underlying calls (keyword search + facet filter) and hands you a clean list of results immediately. It’s not just faster; it's smarter.

Search & Discovery Tools

You don't have to write code or manage multiple API keys to check if a category page is displaying items correctly. You simply ask the agent, 'Check the Home Decor section for trending vases.' The system uses `search_category` and `search_trending` automatically.

The difference now isn't just speed; it’s scope. Your AI client acts as an expert e-commerce consultant, allowing you to audit merchandising rules, test recommendation logic, and monitor market trends without ever touching a developer console.

Klevu (E-commerce AI Search) - 10 Tools

These tools allow you to run every kind of search needed for an e-commerce site, from simple keyword lookups to complex JSON payloads.

#	TOOL	DESCRIPTION
01	`search_autocomplete`	Provides instant suggestions for users as they begin typing into a search bar.
02	`search_category`	Retrieves product listings specifically configured for a given category page structure.
03	`search_filtered`	Narrows down the catalog results by applying multiple specific criteria like color, size, and brand simultaneously.
04	`search_keyword`	Searches the entire product catalog using a general keyword provided in plain text.
05	`search_pagination`	Gets chunks of search results when you need to view more items on long result pages.
06	`search_product_id`	Fetches the complete details for a single product using its unique catalog ID number.
07	`search_raw`	Allows you to send custom, complex JSON payloads directly against the Klevu API endpoints.
08	`search_recs`	Retrieves product suggestions based on machine learning models that predict what users might want.
09	`search_sorted`	Performs a keyword search but allows you to specify how the results should be ordered (e.g., by price or date).
10	`search_trending`	Shows the currently most popular and relevant products across your entire store.

See It in Action

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

U Search for 'waterproof jackets' in my Klevu catalog

Searching catalog... I've found 12 waterproof jackets. Top results include the 'Summit Pro Hardshell', 'RainGuard Trail Jacket', and 'Urban Explorer Parka'. Prices range from $89 to $245. Would you like to filter these by size or brand?

U Show me trending products for the 'Home Decor' category

Retrieving trending items for 'Home Decor'... Current top sellers include 'Minimalist Ceramic Vase', 'Boho Woven Throw', and 'Smart Ambient Lamp'. These items are showing high engagement over the last 24 hours. Would you like to see visual similar items?

U Execute a filtered search for 'sneakers' with brand 'Nike'

Filtering search for Nike sneakers... Found 8 matches. Featured models: 'Air Max Pulse', 'Dunk Low Retro', and 'Zoom Fly 5'. Would you like me to sort these by price from lowest to highest?

Frequently Asked Questions

01 **How do I check if my product catalog supports complex filtering using Klevu AI Search MCP?**

You use the `search_filtered` tool. You just tell your agent what facets you want to combine, like 'color and size,' and it runs the query for you.

02 **Can I find out what products are selling well right now using Klevu AI Search MCP?**

Yes. Use `search_trending`. This tool shows current top sellers, letting you monitor global product velocity and spot seasonal spikes instantly.

03 **What if I need to run a query that the simple tools don't cover? Does Klevu AI Search MCP help?**

Absolutely. If your needs are highly specific, use `search_raw`. This tool lets you execute custom JSON search payloads against any deeply nested part of the Klevu API.

04 **How do I get product suggestions for a user on my site?**

You run `search_recs`. The agent uses this to fetch predictions based on machine learning, giving you suggested items like 'frequently bought together'.

05 **Does Klevu AI Search MCP handle product IDs for single lookups?**

Yes. If you know the ID number of a product, use `search_product_id`. This quickly retrieves all details for that single catalog item.

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 `"klevu-e-commerce-ai-search": { "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

Klevu (E-commerce AI Search) 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 Klevu (E-commerce AI Search). 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	Klevu (E-commerce AI Search) MCP
Server ID	`019d75c1-b49e-72ce-854c-7e583d4d7e66`
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/klevu-e-commerce-ai-search.