MCP SERVER | NO CODE | CLOUD HOSTED

Nabis MCP

Manage wholesale orders, inventory, and compliance data

Nabis connects your AI agent directly to wholesale and distribution data for the cannabis industry. Check real-time stock levels across multiple warehouses, track every order from submission through fulfillment, and pull detailed financial records like invoices and aging reports instantly.

A+ Quality Score 100/100

wholesale-distribution | order-management | inventory-tracking | logistics | cannabis-industry

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

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.

Nabis MCP

10 tools available
Cloud-hosted on Vinkius

This MCP gives your AI client eyes on the entire cannabis supply chain, connecting it to the Nabis Platform V2 API. You stop checking separate dashboards for orders, inventory counts, and payment statuses. Instead, you ask your agent a question—like 'What's our stock of Blue Dream in NY compared to CA?'—and it pulls the answer instantly. Whether you're managing retailer compliance, tracking incoming shipments, or reconciling complex invoices, this MCP handles the raw data retrieval so you don't have to. Everything runs through Vinkius, making Nabis one connection point for all your wholesale needs. It lets you manage orders, check inventory quantities, and review financial reports without ever leaving your chat window.

Core Capabilities

01 — Track Stock Levels

Get real-time stock counts for products across multiple warehouse locations.

02 — Manage Orders

List and pull full details on wholesale orders, allowing you to track their status from start to finish.

03 — Review Financial History

Access detailed records for invoices and payment statuses to monitor your cash flow.

04 — Check Retailer Compliance

Retrieve specific details about individual retailers, including their current license status.

05 — Get Logistics Status

Find out the location of warehouses and check for any company holidays affecting operations.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and provide your Nabis Access Token from their developer portal.

02 Connect your AI client (Claude, Cursor, or another agent) to the catalog. The connection authorizes access using the token you provided.

03 Use natural language prompts within your chat interface; your agent calls the necessary tools—like listing orders or checking inventory—and reports the findings directly back to you.

The bottom line is that instead of logging into five different systems, you talk to one place and get all the data compiled for you.

Built For

This is for operations managers or finance coordinators who are tired of spending their mornings jumping between multiple proprietary dashboards just to reconcile a single shipment. If your job involves coordinating complex, multi-stage wholesale logistics in the cannabis space, this MCP saves you hours.

Operations Manager

Uses this MCP daily to check current stock counts using list_inventory and coordinate delivery schedules by listing warehouses.

Finance Coordinator

Pulls invoice details via get_invoice and lists all outstanding invoices to verify payment statuses against expected revenue.

Wholesale Sales Director

Checks the status of pending orders using list_orders and validates retailer compliance by calling get_retailer for new accounts.

What Changes When You Connect

01 Stop manually checking multiple systems. Your agent pulls together all order details (using get_order) and financial status in one conversational query.

02 Maintain real-time visibility into your stock. Instead of guessing, you can list inventory using list_inventory or check a specific location with get_warehouse_count.

03 Simplify compliance checks. You instantly pull retailer information via get_retailer, ensuring every partner meets the necessary state guidelines before shipping.

04 Streamline accounting tasks. Quickly review past transactions by calling list_invoices to see payment history and reconcile aging reports.

05 Understand logistics at a glance. List warehouses gives you an overview of your physical network while get_days_off ensures you plan around potential operational downtime.

Real-World Applications

Reconciling Payments for the Month

A finance coordinator needs to know which invoices are overdue. They ask their agent, 'Show me all outstanding payments.' The agent uses list_invoices and get_invoice, generating a clear report of who owes what and when.

Verifying Stock for a New Deal

A sales director needs to confirm if enough product exists before committing to a large contract. They prompt the agent with 'What is the current stock level for Product X across all warehouses?' The agent uses list_inventory and get_warehouse_count.

Investigating an Order Delay

The operations team gets flagged about a late shipment. They ask, 'Tell me everything on order #123.' The agent calls get_order to pull the complete history, identifying the exact bottleneck.

Onboarding a New Retail Partner

A compliance officer needs to vet a new location. They ask the agent to check 'Retailer details for ABC Corp.' The agent uses get_retailer and list_retailers to verify license status instantly.

Patterns to Avoid

Treating it like a general database query

✗ AVOID

Trying to ask the MCP, 'What is the average profit margin for all cannabis products?' The Nabis data focuses on logistics and transactions, not deep profitability analysis.

✓ INSTEAD

Stick to transactional queries. If you need financial summaries, use list_invoices or get_invoice. If you need inventory movement, use list_inventory.

Ignoring the time element

✗ AVOID

Assuming all orders are processed on the same day and trying to calculate a rolling average without filtering dates.

✓ INSTEAD

Use the specific tools like get_order or list_orders, and always specify date ranges in your prompt to ensure you're working with accurate historical data.

Relying on one single source of truth

✗ AVOID

Assuming that just because an order is listed (list_orders), the payment has cleared.

✓ INSTEAD

Always cross-reference. Use list_orders to see the commitment, then call get_invoice or list_invoices to verify the financial status.

The Right Fit

Use this MCP if your primary workflow involves managing the physical movement of product and tracking associated money in the cannabis wholesale space. Specifically, if you need to connect order lifecycle management (list_orders), inventory accounting (list_inventory), and payment verification (get_invoice). Don't use it if your needs are purely internal—for example, if you only need to manage employee PTO or HR data; for those, a dedicated HR MCP is better. Also, don't expect predictive analytics on market trends. This tool provides structured historical and real-time operational data (like get_warehouse_count), but it won't tell you what will happen next. It tells you exactly what *is* happening now.

The constant switching between tabs to manage wholesale shipments is exhausting.

Today, managing a single shipment means opening the order portal, jumping over to the inventory dashboard, logging into accounting software for invoices, and finally checking a separate compliance sheet. You spend fifteen minutes just clicking through different systems—copying order IDs here, checking stock levels there—before you even start solving the actual problem.

With this MCP, your agent handles it all in one conversation. Instead of opening five tabs, you ask for an update. It pulls together the full picture: confirming the item is available (list_inventory), that the retailer is compliant (get_retailer), and that the payment was already processed (get_invoice). You get a single, actionable answer.

Nabis MCP gives you immediate visibility into your entire operational status.

The tedious manual steps—like cross-referencing the shipping address from an order (get_order) against the current warehouse location (list_warehouses)—are eliminated. You just ask, and it checks both sources for you.

What's different now is that you treat your entire supply chain as a single data source accessible via natural conversation. It cuts out the friction between 'where is it?' and 'has it been paid for?'

Nabis: 10 Tools for Supply Chain Management

Use these tools to pull specific data points about orders, stock levels, invoices, and retailers directly into your conversation.

#	TOOL	DESCRIPTION
01	`get_days_off`	Retrieves specific dates when Nabis is not operating, helping you plan around company closures.
02	`get_invoice`	Pulls a detailed record for one specific invoice number.
03	`get_order`	Fetches all the details related to a single, specified wholesale order.
04	`get_retailer`	Retrieves comprehensive data points about an individual licensed retailer.
05	`get_warehouse_count`	Gets the current stock quantity available at a specific warehouse location.
06	`list_inventory`	Generates a list of all product inventory quantities across your network.
07	`list_invoices`	Lists multiple invoice records, allowing you to see payment summaries and aging reports.
08	`list_orders`	Retrieves a summary list of all wholesale orders placed through the Nabis platform.
09	`list_retailers`	Provides a searchable list of all registered retailers in your network.
10	`list_warehouses`	Lists every active warehouse location, including their physical addresses and status.

See It in Action

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

U List all pending Nabis orders.

Fetching orders... I've found 8 pending orders. Here are the most recent ones...

U What is the stock level for product 'Blue Dream' in the LA warehouse?

Checking inventory... 'Blue Dream' currently has 150 units available in the Los Angeles warehouse.

Frequently Asked Questions

01 **How do I check if Nabis is closed when I need to list_orders?**

You should first use get_days_off. This tool confirms if there are any company holidays or scheduled closures, preventing you from running reports that will fail due to system downtime.

02 **Can Nabis MCP tell me the profit margin on an order?**

No, this MCP focuses on logistics and transactions. It can pull all the raw data—like getting_order details or list_invoices—but you'll need a separate analytics tool for calculating margins.

03 **Is get_warehouse_count reliable for real-time stock?**

Yes, this tool provides current stock counts from your designated warehouses. Always cross-reference with list_inventory if you need a comprehensive view across all product lines.

04 **Do I need to use list_retailers before getting_retailer data?**

No, they serve different purposes. list_retailers gives you a directory of every partner. get_retailer pulls deep details about one specific partner when you know their name.

05 **What is the difference between list_orders and get_order?**

list_orders provides a summary view of all orders, letting you see who placed an order. get_order pulls every single piece of data for one specific order ID.

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 `"nabis": { "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

Nabis 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 Nabis. 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	Nabis MCP
Server ID	`019d75da-2a40-71ed-9a1a-ccf2315824de`
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/nabis.