

MCP SERVER

NO CODE

CLOUD HOSTED

ShipBob MCP

Manage Inventory, Orders, and Fulfillment Logistics

ShipBob MCP connects your AI agent directly to your e-commerce fulfillment system, giving you full control over your supply chain. You can manage product records, track inventory across multiple locations, create new orders, and update shipment details—all through natural conversation. It eliminates the need to jump between your store backend and the ShipBob dashboard just to check stock or cancel a shipment.

A+ Quality Score 98.33/100

fulfillment

order-tracking

supply-chain

warehouse-management

logistics



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

ShipBob MCP

30 tools available

Cloud-hosted on Vinkius

Managing e-commerce logistics shouldn't require three different browser tabs and a dedicated operations degree. This MCP connects your AI agent directly to ShipBob, letting you run your entire fulfillment process through conversation. Instead of navigating complex dashboards, you simply ask for what you need: 'What's the stock level for SKU 123?' or 'Can we cancel order 500?'

This tool lets you handle everything from managing product variants and converting items into bundles to tracking down historical inventory movements. If you're looking for a way to put all your supply chain data in one place, check out the Vinkius catalog; connecting this MCP gives your AI client immediate access to deep operational capabilities.

It's about taking full command of order processing and warehousing visibility. You can create return orders, manage warehouse receiving documents, or just get a detailed status on any shipment without ever leaving your chat interface.

Core Capabilities

01 — Track Orders and Shipments

Get real-time details, current statuses, and tracking numbers for specific fulfillment orders.

03 — Create and Modify Products

Establish new products, manage product variants, or combine existing items into bundles.

05 — Update Logistics Data

Correct shipping addresses, cancel multiple shipments at once, or update tracking information.

02 — Manage Inventory Levels

Check stock counts by location, group lot, or get an aggregated overview of all available inventory units.

04 — Handle Returns and Receiving

Generate return merchandise authorizations (RMAs) or create warehouse receiving orders (WROs).

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to the MCP and enter your ShipBob API Token.
- 02 Connect this MCP to your preferred AI client (Claude, Cursor, etc.).
- 03 Ask your agent a question or issue a command—for example, 'Show me all inventory levels grouped by lot'—and get instant data back.

The bottom line is you stop jumping between systems and manage logistics directly through natural conversation with your AI client.

Built For

Anyone managing e-commerce inventory or fulfillment runs into the same bottleneck: too much data spread across too many dashboards. This MCP is for the operations manager who needs visibility, and the customer service agent who can't afford to spend minutes tracking down a single order number.

E-commerce Owner

Checks stock levels or creates new products instantly without logging into multiple administrative portals.

Customer Support Agent

Retrieves tracking numbers and updates shipping addresses for customers in seconds to resolve inquiries quickly.

Operations Manager

Monitors warehouse receiving activity, checks inventory history, or cancels bulk shipments when supply chain plans change.

What Changes When You Connect

- 01 Stop bouncing between dashboards. Your AI agent handles complex tasks like generating a return order (`create_return`) or updating an address on the fly.

-
- 02** Never guess your stock count again. You can get accurate inventory levels by location using `get_inventory_levels_by_location` , so you always know what's available to ship.
-
- 03** Product maintenance is simple. Instead of manual data entry, you can create new products (`create_product`) or combine variants into bundles (`convert_variant_to_bundle`) with a single prompt.
-
- 04** Customer service improves instantly. If a customer needs an address change, your agent runs `update_shipment_address` immediately, confirming the fix without manual clicks.
-
- 05** Audit trails are clean. You can query inventory history using `query_inventory_history` and review all transaction records via `query_transactions` , giving you total accountability.
-
- 06** Speed up receiving. Instead of printing labels manually, you request them with `get_receiving_labels` , getting the required PDF box labels instantly.
-

Real-World Applications

A shipment needs an address change before dispatch.

The CS agent realizes the customer gave the wrong zip code. Instead of emailing the ops team to manually correct it, they use their AI client to run ``update_shipment_address``, fixing the error instantly and confirming the new delivery details.

A batch of orders must be canceled due to supplier delays.

Instead of going into the dashboard and canceling 20 separate orders one by one, the manager uses ``batch_cancel_shipments`` with a simple prompt, clearing out all affected shipments in bulk.

Inventory count is off across different warehouses.

The Ops Manager needs to know if a specific SKU is available at three different fulfillment centers. They prompt the agent to run ``get_inventory_levels_by_location`` and get immediate, segmented stock numbers.

A customer returns an item that needs to be restocked.

The agent initiates the process using ``create_return``, logs the incoming goods as a Warehouse Receiving Order (``create_receiving``), and then uses ``list_returns`` to track its progress through the system.

Patterns to Avoid

Searching multiple dashboards for stock.**✗ AVOID**

The team spends 15 minutes logged into three different systems (inventory, orders, and product catalog) trying to verify if an item is ready to ship, only finding out at the end that two sites are out of sync.

✓ INSTEAD

Use your AI client with this MCP. Ask it to ``get_inventory_levels`` or ``list_products``. The agent pulls all three data points for you in one query.

Manually correcting shipment addresses.**✗ AVOID**

A customer calls about a wrong address, and the support rep has to wait 10 minutes while an ops teammate manually logs into another portal just to make the change.

✓ INSTEAD

The agent runs ``update_shipment_address``. It performs the correction instantly through conversation, notifying the user right away.

Overlooking historical inventory movements.**✗ AVOID**

Management needs to know why stock dipped last month. They have to download multiple reports and compare spreadsheets just to piece together the movement timeline.

✓ INSTEAD

Use ``query_inventory_history``. You get a single, coherent report of every movement event in one place.

The Right Fit

Use this MCP if your primary pain point is data fragmentation across multiple e-commerce operational dashboards. If you frequently need to check stock levels by lot/expiration date (`get_inventory_levels_by_lot`), or if managing the lifecycle of an order (from `create_order` to `update_shipment_address`) involves jumping between systems, this is essential. You'll love it.

Don't use this MCP if you just need general market intelligence or deep accounting analysis that doesn't touch fulfillment data. If your task is simply listing available fee types (`list_transaction_fees`), a simple database query tool might suffice and be less complex than the full scope of the ShipBob tools.

The e-commerce dashboard overload is exhausting.

Today, managing fulfillment means toggling between your storefront backend, the dedicated inventory system, the order management platform, and the shipping carrier portal. You copy an SKU from one tab to another, you paste a tracking number into a third, then switch back out just to check if that item has been packed. It's slow, it's error-prone, and frankly, it takes up too much time.

With this MCP, your AI agent acts as the central hub for all that data. You simply ask: 'What is the status of order 900?' The response includes the order details, the current shipment status from `get_order` , and confirmation on whether the address needs an update via `update_shipment_address` . It brings everything together.

ShipBob MCP: Full Command Over Your Supply Chain

Manual tasks like creating a return order (`create_return`) or setting up warehouse receiving records are often slow, requiring multiple clicks and status checks across different pages. This means delayed restocking and lost time.

Now, you ask your agent to `create_receiving` or `cancel_order` . The MCP handles the multi-step process behind the scenes. You get confirmation, instantly. It's done.

ShipBob: 30 Tools for Fulfillment & Logistics

Use these tools to manage everything from order cancellation and address updates to detailed inventory history queries, all within your AI chat interface.

#	TOOL	DESCRIPTION
01	<code>batch_cancel_shipments</code>	Cancels an entire group of shipments simultaneously.
02	<code>batch_update_tracking</code>	Marks multiple tracking records as successfully synced to your internal system.
03	<code>cancel_order</code>	Cancels a specific order and any associated shipments.
04	<code>list_channels</code>	Retrieves a list of all e-commerce sales channels the account is connected to, such as Shopify or Amazon.
05	<code>convert_variant_to_bundle</code>	Combines product variants into a single new bundle listing.
06	<code>create_order</code>	Generates and submits a new fulfillment order to the warehouse.
07	<code>create_product</code>	Sets up entirely new products, including all required variants.
08	<code>create_receiving</code>	Generates a formal Warehouse Receiving Order (WRO) for incoming inventory.
09	<code>create_return</code>	Initializes and manages a return merchandise authorization (RMA).
10	<code>get_inventory_levels_by_location</code>	Retrieves current inventory stock counts, separated by specific fulfillment center locations.
11	<code>get_inventory_levels_by_lot</code>	Checks how much inventory is available, grouped specifically by lot or expiration date.
12	<code>get_inventory_levels</code>	Gets a comprehensive total count of all inventory items across every location.
13	<code>get_order</code>	Fetches detailed information and the current shipping status for any given order number.
14	<code>get_product</code>	Retrieves all details associated with a specific product listing.
15	<code>get_receiving_labels</code>	Downloads printable PDF box labels required for a Warehouse Receiving Order (WRO).

#	TOOL	DESCRIPTION
16	<code>get_shipment</code>	Retrieves specific, granular details about an already created shipment.
17	<code>list_inventory</code>	Lists all inventory items currently tracked in the system.
18	<code>list_invoices</code>	Retrieves a list of invoices based on specified date ranges.
19	<code>list_orders</code>	Provides a searchable list of orders, allowing filtering by status or tracking information.
20	<code>list_products</code>	Lists all available products in the catalog with advanced filtering options.
21	<code>list_receiving</code>	Shows a list of all active Warehouse Receiving Orders (WROs).
22	<code>list_returns</code>	Displays a searchable list of all return orders.
23	<code>list_transaction_fees</code>	Lists the different types of transaction fees that apply to sales.
24	<code>list_product_variants</code>	Gets every size, color, or style variation associated with a particular product.
25	<code>query_inventory_history</code>	Queries and reports on all historical inventory movement events for auditing purposes.
26	<code>query_transactions</code>	Searches through financial records, including charges and credits.
27	<code>set_external_sync_receiving</code>	Marks the Warehouse Receiving Orders (WROs) as having been successfully updated in your external system.
28	<code>update_product</code>	Changes and saves modified details for an existing product listing.
29	<code>update_return</code>	Modifies the status or details of a return order that is already active.
30	<code>update_shipment_address</code>	Corrects and updates the final shipping address for an existing shipment.

See It in Action

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

U List all my active ShipBob sales channels.



I've retrieved your channels. You have 2 active installations: 'Shopify Store Front' (ID: 4567) and 'Amazon Integration' (ID: 8910).

U What is the current status and tracking for order 123456?



Order 123456 is currently 'Shipped'. The tracking number is 1Z999AA10123456789 via UPS. It was dispatched from the Chicago fulfillment center.

U Show me the inventory levels for SKU 'SUMMER-HAT-01'.



For SKU 'SUMMER-HAT-01', you have 150 units fulfillable across all locations, with 25 units currently committed to open orders.

Frequently Asked Questions

01 How do I check inventory levels with ShipBob MCP?

You can use ``get_inventory_levels``. This tool lets you see the total stock count across all locations, or if you need more detail, you can specify if you want to view stock grouped by lot using ``get_inventory_levels_by_lot``.

02 Can I update a shipping address with ShipBob MCP?

Yes. Use the ``update_shipment_address`` tool. It lets your agent correct and save a new shipping location for an existing shipment, preventing costly delays.

03 What is the best way to handle returns with ShipBob MCP?

Start by using `create_return`. This generates the necessary return order (RMA) in the system. After the item arrives, you can then manage its intake using `list_returns`.

04 Does ShipBob MCP let me create new products?

Yes. You use `create_product` to set up brand new items and their variants. If you have existing product variations that need grouping, `convert_variant_to_bundle` handles that for you.

05 How do I get a list of my active sales channels?







Simply ask the agent to run `list_channels`. It will pull all connected e-commerce platforms, like your Shopify or Amazon integrations, into one easy response.

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 <code>"shipbob": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

ShipBob 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](https://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 ShipBob. 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	ShipBob MCP
Server ID	019e38eb-bb03-7395-9e9a-708473493c83
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/shipbob.