MCP SERVER | NO CODE | CLOUD HOSTED

GPTBots MCP

Test, Audit, and Control Your Agent Infrastructure

GPTBots lets you manage your entire conversational AI infrastructure directly from your development environment. It gives developers and ops teams direct access to test bot responses, review chat histories, upload knowledge documents, and trigger complex automated workflows—all through a single API connection.

A+ Quality Score 100/100

chatbot | workflow-automation | knowledge-base | conversational-ai | bot-management

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

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.

GPTBots MCP

8 tools available
Cloud-hosted on Vinkius

You connect this MCP to your agent client, giving it full control over your enterprise AI setup. You can interact with deployed bots by sending messages or listing past conversations. If you need to audit performance, check the chat history of any bot at any time. It's also how you manage the data powering those bots: list available knowledge documents and upload new content directly to keep the context fresh. Need automation? Trigger configured AI workflows programmatically and even query their execution status. This MCP makes your agent reliable for real-world use, which is exactly what Vinkius delivers by hosting this connection in one place.

Core Capabilities

01 — Manage Bot Conversations

View a list of active chats and retrieve the full chat history between a user and an AI agent.

02 — Update Knowledge Bases

Upload new documents or view existing content to keep your bot's contextual information current.

03 — Orchestrate Workflows

Start complex automated processes and check the status of those executions without manually clicking through a dashboard.

04 — Inspect Data Sources

List available tables and records hosted within your platform database for data queries.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius.

02 Provide your GPTBots API Key and Data Center Region credentials.

03 Connect everything via your AI client, then use the tools to manage agents and workflows.

The bottom line is you get a single connection point into complex bot management, testing, and data pipelines.

Built For

This MCP targets technical roles who build and maintain conversational AI products. It's for the developer stuck in their IDE who can't afford to leave their coding environment just to test a bot's knowledge base, or the operations lead who needs to audit conversations across multiple platforms.

AI Developer

Tests agent responses and updates knowledge bases directly from the IDE without needing a separate UI.

Product Manager

Audits conversation histories to evaluate bot performance, spot common failure points, and assess user satisfaction.

Operations Engineer

Integrates automated workflows into existing daily processes and manages data records programmatically for reliability checks.

What Changes When You Connect

01 You get instant access to chat history. Instead of logging into a separate dashboard to see what happened last week, you can use list_conversations and then get_conversation to pull up exact transcripts instantly.

02 Knowledge maintenance is streamlined. You don't have to manually upload files via a web portal; you just call create_knowledge_document from your code to keep the bot current.

03 Workflow debugging gets easier. If an automated process fails, you can't wait for an email alert. Use trigger_workflow and then query_workflow to check its status right away.

04 Data visibility is key. By calling list_databases, you immediately see what data sources your bots rely on, which is crucial before writing any code.

05 Testing agents is faster than ever. Instead of waiting for a manual test run, you can send_bot_message to simulate user input and check the response in real time.

Real-World Applications

Auditing Bot Performance After an Incident

An Ops team member notices bot performance dropped after a software update. They use list_conversations to pull up chats from the last hour, then call get_conversation on specific IDs to compare chat content before and after the drop, quickly pinpointing where the knowledge context failed.

Onboarding a New Knowledge Source

A Product Manager receives 50 new legal documents. Instead of manually uploading them one by one, they use create_knowledge_document to bulk-upload all files into the knowledge base, ensuring the bot is instantly updated for compliance questions.

Testing Automated Business Flows

A developer needs to test a user onboarding process. They use trigger_workflow to start the process and pass required parameters. Once initiated, they follow up with query_workflow to ensure every step finished successfully before marking it 'live'.

Building Internal Bot Debugging Tools

A developer wants an internal script that verifies agent dependencies. They first list_databases to see all available data tables, then use send_bot_message to test the bot's ability to reference specific records from those tables.

Patterns to Avoid

Treating it like a simple chat UI

✗ AVOID

Trying to 'talk' to the MCP through your agent client as if it were a chatbot. You might try asking, 'What documents do I have?' and expect conversational answers.

✓ INSTEAD

Use dedicated functions instead of natural language queries. To see what knowledge is available, call list_knowledge_documents. To check chat history, you must use the get_conversation tool.

Ignoring workflow status

✗ AVOID

Triggering a complex billing process using trigger_workflow and then assuming it finished instantly. You might write code that relies on the outcome before verification.

✓ INSTEAD

Always follow up by calling query_workflow with the execution ID returned from triggering the flow. This confirms success or failure before proceeding.

Hardcoding credentials

✗ AVOID

Writing your API key directly into the client code instead of letting the MCP handle authentication.

✓ INSTEAD

Use this MCP through Vinkius's secure connection method. This keeps your sensitive keys managed and accessed only by authorized agents.

The Right Fit

Use this MCP if you need to treat conversational AI agents like backend services—if testing, auditing, or managing the agent is a programmatic necessity, use this. You must be able to write code that initiates actions (like trigger_workflow) or retrieves structured data (like list_databases). Don't use it if your only goal is casual conversation; for simple chat interfaces, you just need a basic messaging connector. If your primary need is simply generating text from a prompt without checking the state of knowledge documents or workflows, another pure LLM wrapper might suffice. But since you manage enterprise data and complex processes, this MCP gives you the necessary control plane.

Manually tracking bot performance across multiple systems is hellish.

Right now, if a user complains that your AI agent gave bad advice, you have to manually hop into three different dashboards: the chat log viewer (to see *what* was said), the knowledge management portal (to check *if* it knew the right thing), and then maybe even a workflow console (to see *why* it failed). You copy IDs here, paste them there. It's slow, it's error-prone, and you spend half your day being an auditor instead of building.

With this MCP connection, you can pull all that data into one place via your agent client. You don't jump between tabs anymore. You simply ask your agent to list_conversations, then get the full history using get_conversation. The entire audit trail is exposed programmatically.

Accessing Knowledge and Workflows with GPTBots

The biggest manual step that vanishes is the need to guess what data sources your bot can use or how a process runs. You don't have to hunt through documentation pages to see what tables exist; you just run list_databases. And instead of manually restarting processes, you call trigger_workflow and immediately check status with query_workflow.

It changes the game from reactive troubleshooting to proactive engineering. Your bot isn't a black box anymore; it’s an auditable, controllable system built directly into your development cycle.

GPTBots: 8 Tools for Agent Control

These eight tools give you granular control over every part of your AI agent infrastructure, from data querying to triggering complex automation.

#	TOOL	DESCRIPTION
01	`list_databases`	Reads the names of all tables hosted in your platform database.
02	`create_knowledge_document`	Allows you to upload new files or create documents within the knowledge base.
03	`get_conversation`	Retrieves specific details and the full chat transcript for a single conversation.
04	`list_conversations`	Fetches a list of all past chats that occurred with your bots.
05	`list_knowledge_documents`	Shows you the titles and metadata of documents currently stored in the knowledge base.
06	`query_workflow`	Checks if an automated workflow has finished running, providing its execution status.
07	`send_bot_message`	Sends a direct message to one of your deployed AI agents for testing or interaction.
08	`trigger_workflow`	Starts an automated, pre-configured workflow sequence immediately.

See It in Action

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

U List all recent conversations for bot ID 'bot_123xyz'.

Fetching conversations... I found 3 active conversations for this bot. Conversation ID 'conv_987' was updated 5 minutes ago. Would you like to see its chat history?

U Trigger the onboarding workflow (ID: 'wf_456') and pass the parameter email='test@example.com'.

Triggering workflow... Success! The workflow has been initiated. The execution Record ID is 'rec_789abc'. Let me know if you want me to check its status.

Frequently Asked Questions

01 **How do I check if my automated workflow ran successfully using GPTBots MCP?**

You use the query_workflow tool. First, you call trigger_workflow to start the process, and that action returns a record ID. You then pass this ID into query_workflow to see its current status (running, successful, or failed).

02 **Does GPTBots MCP let me upload new documents for my bot?**

Yes. Use the create_knowledge_document tool. This lets you programmatically add new files or update existing knowledge bases to keep your agent's context current.

03 **What is the difference between list_conversations and get_conversation in GPTBots MCP?**

list_conversations gives you a high-level summary, showing all recent chats that occurred. You need to use get_conversation, providing a specific conversation ID, if you want to retrieve the full message history.

04 **Can I test my bot's responses without using a web browser?**

Absolutely. You can send_bot_message directly from your agent client via this MCP. This lets developers test interactions and see immediate responses right within their IDE.

05 **Is the data I need to query available through list_databases in GPTBots MCP?**

The tool simply lists all tables hosted on the platform database. You must then use your agent client's capability to query records within those specific tables.

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 `"gptbots": { "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

GPTBots 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 GPTBots. 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	GPTBots MCP
Server ID	`019d75aa-3c45-70d3-8172-0f8f706b203f`
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/gptbots.