

MCP SERVER

NO CODE

CLOUD HOSTED

# Freshdesk MCP

Coordinate support operations via conversation.

Freshdesk MCP gives your AI agent full control over your customer support ecosystem. Instantly manage tickets, look up contact histories, check internal team status, and update ticket statuses—all from a natural conversation. You get an operational coordinator that lives inside your favorite client.

**A+** Quality Score 100/100

ticket-management

omnichannel-support

customer-service

automation

self-service

workflow-management



# 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.

### 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](https://cloud.vinkius.com) — connect your AI agent in under 60 seconds.

# Freshdesk MCP

10 tools available

Cloud-hosted on Vinkius

This connection lets you turn your helpdesk into something truly automated. Instead of jumping between tabs to check if a ticket is open or who the customer is, you just talk to your agent. It acts like having a dedicated operations manager sitting next to you.

Need to know the background on an issue? Your agent pulls up detailed contact profiles and full interaction histories immediately. If a ticket status is wrong, you can update it—changing priorities or moving it from 'Open' to 'Pending' without logging into Freshdesk itself. You can also list all agents and groups to figure out who needs to handle the request next. When you connect this MCP through Vinkius, your AI client becomes a central command center for everything support-related.

---

## Core Capabilities

### 01 — Retrieve core customer data

Pull complete profiles and interaction histories for specific contacts or entire organizations.

### 02 — Manage ticket lifecycle

List existing tickets, get full metadata on any case, create new requests, or change the status/priority of a ticket.

### 03 — Map out internal support structure

Access lists of active support agents, available groups, and knowledge base folders to coordinate team assignments.

# One Click on Vinkius — From Prompt to Execution

Available at [vinkius.com/mcp/freshdesk-alternative](https://vinkius.com/mcp/freshdesk-alternative) — connect your AI agent in three steps.

- 01 Subscribe to this MCP and grab your API Key from Freshdesk's profile settings.
- 02 Identify your specific account subdomain (like 'vinkius' in [vinkius.freshdesk.com](https://vinkius.freshdesk.com)).
- 03 Connect the credentials within any MCP-compatible client, giving your agent immediate access to your support data.

The bottom line is that once it's connected, you never have to manually log into Freshdesk again just to get an overview or make a quick update.

---

## Built For

This MCP is for the Ops Lead who gets frustrated having to jump between five different screens just to triage one ticket. It's built for support teams drowning in manual status checks, and Customer Success Managers who need instant access to interaction history without leaving their primary workspace.

### Support Operations Lead

Manages agent groups and monitors team workload by listing available staff and coordinating ticket routing via natural language commands.

### Customer Success Manager

Retrieves full interaction histories for contacts, verifies account details, and documents the outcome of a conversation without needing to open the main helpdesk UI.

### Tier 2 Support Agent

Quickly checks ticket metadata, reviews knowledge base folders, and updates ticket statuses or priorities when escalating an issue.

---

## What Changes When You Connect

- 01 Stop manual status checks. Use the `update_ticket_details` tool to change a ticket's priority or status instantly, regardless of which tab you're currently viewing.

- 
- 02 Get instant context on customers. The agent can run `get_contact_details` and immediately pull up full interaction histories for any contact ID you provide.

---

  - 03 Streamline new issue intake. Don't waste time writing emails to open a case; just ask the agent to `create_ticket`, and it handles all the required fields.

---

  - 04 Know who's available. Use `list_support_agents` or `list_support_groups` to check team availability before assigning an issue, improving your internal routing process.

---

  - 05 Quickly search knowledge. Instead of browsing folders manually, you can use tools like `list_help_folders` to map out the support documentation structure and find context faster.
- 

---

## Real-World Applications

### **Triage a high-priority billing issue.**

A customer reports a sudden billing problem. Instead of manually checking the ticket queue, you ask your agent to `list_tickets` and filter by 'Billing'. You grab the full conversation thread using `get_ticket_details`, confirm it's High priority, and then use `update_ticket_details` to assign it immediately to the Billing Support Group.

### **Manage a large-scale outage.**

The team discovers an issue affecting multiple users. You ask your agent to list all support agents and groups via `list_support_agents` and `list_support_groups`. This allows you to quickly determine which teams need to be alerted or assigned the incoming tickets.

### **Onboard a new customer account.**

A sales rep closes a deal but needs support documentation. They ask your agent to find all contacts related to the company via `list_support_companies` and then use `get_contact_details` for the primary user, ensuring all necessary information is logged before handing off the case.

### **Document a feature suggestion.**

A customer calls with an idea for improvement. Instead of writing an email, you ask your agent to `create_ticket` and specify 'Feature Suggestion' as the subject, ensuring it gets logged into the official system immediately.

---

# Patterns to Avoid

---

## Copying ticket IDs manually

### ✗ AVOID

A user copies a ticket ID from an email, pastes it into another window, and then has to remember which field in the helpdesk UI corresponds to that specific metadata.

### ✓ INSTEAD

Just ask your agent for ``get_ticket_details`` using the ID. It pulls all the necessary data—metadata, threads, etc.—into one conversational response without you touching a single form field.

---

## Forgetting current status

### ✗ AVOID

A support agent thinks a ticket was resolved yesterday but can't remember if the 'Resolved' status actually moved it to the queue for final review.

### ✓ INSTEAD

Use ``get_ticket_details`` first. It shows you the full audit log and current status, letting you confirm exactly what needs to happen before running ``update_ticket_details``.

---

## Relying on email updates

### ✗ AVOID

A customer sends a follow-up via email that requires manual creation of a new ticket record and assignment to the correct group.

### ✓ INSTEAD

Tell your agent to ``create_ticket`` for you. You just need to provide the subject and contact, and it handles the entire submission process into Freshdesk.

---

## The Right Fit

Use this MCP if your biggest workflow pain point is context switching. If you constantly juggle between your chat client, the ticket queue, and internal team directories, this is for you. You need to manage the full lifecycle of a customer interaction—from initial contact (using `create_ticket`) through investigation (`get_contact_details`, `list_tickets`) to final action (`update_ticket_details`). Don't use it if all you need is simple data aggregation, like just listing contacts. For that limited scope, a standalone directory tool might be enough. But because this MCP handles everything from team status checks (`list_support_agents`) to detailed ticket history, it's the central hub for any support operation.

---

---

## The friction of modern helpdesk management

Right now, managing a single customer issue means bouncing between five different places. You check your email for updates, you click into the main ticket queue to see if it's open, you tab over to the contact directory to verify details, and then you finally jump back to the case file just to change the status from 'Pending' to 'Ready for Review'. It's a mess of tabs and copy-pasting.

With this MCP, that whole process collapses into a single conversation. You tell your agent what needs doing—whether it's checking who owns the account or moving a ticket—and you get the answer without leaving your current screen. Your AI acts like an expert pair of hands, executing the steps for you.

---

## Automate ticket creation and status management with Freshdesk MCP

You eliminate the need to manually switch between creating a new record and updating its metadata. Instead of filling out forms or finding the right dropdown menu, you simply instruct your agent to `create_ticket` for a specific user and then immediately follow up by asking it to set the priority using `update_ticket_details`.

The result is that every action—from logging an issue to closing a case—is handled by natural language commands. You stop working *in* the helpdesk software, and you start managing your support workflow *with* AI.

---

# Freshdesk: 10 Tools for Support Automation

These tools let you programmatically manage every aspect of your customer service workflow, from creating new tickets to updating agent statuses.

#	TOOL	DESCRIPTION
01	<code>list_support_agents</code>	Retrieves a list of all support staff members within your helpdesk system.
02	<code>list_support_companies</code>	Gets a directory listing of all companies and organizations in your account.
03	<code>list_contacts</code>	Retrieves a list of individual helpdesk contacts by name or criteria.
04	<code>list_help_folders</code>	Shows the organizational structure and names of your knowledge base folders.
05	<code>list_support_groups</code>	Lists the different support agent groups defined in your helpdesk setup.
06	<code>list_tickets</code>	Retrieves a list of all open and closed tickets, allowing you to filter by status or priority.
07	<code>create_ticket</code>	Generates a brand new support ticket directly from your conversation with the agent.
08	<code>get_contact_details</code>	Fetches specific, detailed profile information for any given contact.
09	<code>get_ticket_details</code>	Pulls all metadata for a specific ticket, including the full conversation thread and audit logs.
10	<code>update_ticket_details</code>	Modifies an existing ticket by changing its status or adjusting its assigned priority level.

---

## 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 open tickets in Freshdesk.



I've retrieved 5 open tickets. Notable requests include 'Login issues' (ID: 101) and 'Billing question' (ID: 102). Would you like the full details for any specific ticket?

### U Show me the details for ticket ID '101'.



Fetching metadata for ticket 101... The issue is reported by 'john@example.com'. It's currently 'Open' with 'High' priority. The latest message says the user can't access the dashboard. Shall I list the full conversation?

### U Create a new ticket 'Feature suggestion' for 'sarah@example.com'.



Done! I've created the new ticket for Sarah. The ID is 103 and I've set the priority to 'Low'. Your team will see this in the Freshdesk queue immediately.

---

## Frequently Asked Questions

### 01 How do I get full customer history using Freshdesk MCP?

You can retrieve a contact's complete interaction history by asking the agent to run ``get_contact_details``. This gives you more than just names; it provides detailed profiles and all recorded interactions in one place.

### 02 Can Freshdesk MCP help me update a ticket status?

Yes, you can. Use the ``update_ticket_details`` tool to modify any existing ticket's status or priority level simply by telling your agent what change is needed.

**03 Does Freshdesk MCP help me find which agents are available?**

Absolutely. You can use the `list\_support\_agents` tool to get a list of all staff, and `list\_support\_groups` helps you understand how your team is structured for better routing.

---

**04 What if I need to create a ticket but don't know the ID?**

No problem. You can simply instruct the agent to `create\_ticket`. It handles generating the new case and submitting it into your live Freshdesk queue for you.

---

**05 Which tools do I use to list all tickets in Freshdesk MCP?**

You'll use the `list\_tickets` tool. This function lets you pull a comprehensive list of existing cases and even allows filtering by status or priority, narrowing down your focus immediately.







---

# Go Live in 60 Seconds

Get your connection token from [cloud.vinkius.com](https://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
 <b>Claude AI</b>	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
 <b>Cursor</b>	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
 <b>VS Code</b>	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add <code>"freshdesk-alternative": { "url": "..." }</code>
 <b>Windsurf</b>	MCP Settings → <code>mcp_settings.json</code> → Add endpoint URL
 <b>ChatGPT</b>	Settings → Tools & plugins → Add MCP server → Paste endpoint
 <b>Gemini</b>	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

# Freshdesk 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](https://vinkius.com) · [support@vinkius.com](mailto: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 Freshdesk. 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	Freshdesk MCP
Server ID	019dd0f4-b132-7237-ab7b-9d674dc53f5e
Platform	Vinkius Cloud for AI Agents
Endpoint	<a href="https://edge.vinkius.com/{token}/mcp">https://edge.vinkius.com/{token}/mcp</a>

### 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/freshdesk-alternative](https://vinkius.com/mcp/freshdesk-alternative).