MCP SERVER | NO CODE | CLOUD HOSTED

Freshservice MCP

Manage your entire IT service stack from conversation.

Freshservice connects your agent to its entire IT Service Management system. Use this MCP to manage tickets, track hardware assets, and handle change requests directly through conversation. Instead of navigating multiple dashboards, you can ask your AI client to list active incidents, check asset metadata, or open a brand new ticket instantly.

A+ Quality Score 100/100

itsm | asset-management | cmdb | incident-management | change-management | it-operations

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

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.

Freshservice MCP

12 tools available
Cloud-hosted on Vinkius

Freshservice brings the full scope of IT Service Management (ITSM) into your conversations with your agent. Whether you're managing day-to-day helpdesk operations or tracking complex infrastructure changes, this MCP lets you interact with all your core data without leaving your primary workspace. You can ask for details on a user's profile and instantly check their open tickets, or list every piece of company hardware in the CMDB. Need to start a new incident? Just tell your agent, and it creates the ticket automatically. All these capabilities are available through Vinkius, giving you access to 4,000+ connections in one place.

This MCP lets you automate tasks that usually require clicking through five different tabs: listing all active tickets, checking related assets, verifying user contacts, or even monitoring change request statuses. It keeps your IT operations data readily available for natural conversation.

Core Capabilities

01 — Create new service desk incidents

Automatically opens a fresh IT support ticket when you tell your agent to do so.

02 — Check current helpdesk status

Verifies the operational status of the service desk system, confirming if tickets can be opened or updated.

03 — Retrieve detailed asset information

Pulls metadata for specific IT hardware or software assets from the CMDB inventory.

04 — Get user and requester details

Fetches full profiles, department assignments, and contact information for any employee or user.

05 — List and check ticket data

Retrieves lists of all open tickets and provides detailed metadata for specific incidents.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius, then input your Freshservice Domain prefix and API Key into your agent's configuration.

02 Your AI client authenticates with the platform, making all ITSM data accessible via a single connection.

03 You simply instruct your agent—for example, 'List all high-priority tickets for Engineering'—and it handles the complex API calls.

The bottom line is that you treat your entire service desk and IT asset database like an extension of your conversation.

Built For

IT Managers, Service Desk Agents, and System Administrators. These are for people whose day involves jumping between ticket dashboards, CMDB spreadsheets, and user directories just to answer a single question.

Service Desk Agent

They use this MCP to get instant details on an incoming ticket or requester profile via simple conversation prompts, saving them from manual dashboard navigation.

IT Manager

Managers rely on it to quickly list change requests or review asset metadata across departments without having to open multiple reports.

System Administrator

They use it to automate the retrieval of release records and monitor problem tickets, keeping track of IT compliance and stability.

What Changes When You Connect

01 Eliminate switching between tabs. Instead of opening the CMDB, then navigating to a user directory, you can ask your agent for asset details and requester profiles in one go.

02 Speed up incident handling dramatically. Use list_itsm_tickets to see all open issues, then get_ticket_details immediately gives you enough context to start working without clicking into the ticket record.

03 Improve compliance visibility. You can monitor change requests using list_change_requests and check related problem records (list_it_problems) to ensure every deployment is documented properly.

04 Faster onboarding for new agents. Agents use list_support_agents and list_requesters to quickly understand who needs help and who is authorized to work on tickets, reducing training time.

05 Simplify complex data retrieval. You can ask the agent to find all assets (list_it_assets) assigned to a specific department or user profile via get_requester_details.

Real-World Applications

A new incident needs immediate filing.

An employee calls about their VPN failing. Instead of manually navigating the ticketing system, your agent simply listens to the complaint and uses create_itsm_ticket to open a high-priority incident immediately, logging all required details from the conversation.

Tracing an asset's history.

A manager needs to know if a laptop (asset ID 456) is covered by a recent software release. They ask their agent; it uses get_asset_details and cross-references list_it_releases, providing a single answer instead of requiring two manual lookups.

Understanding user workload.

You want to check if John Doe has too many open issues. You ask your agent; it uses get_requester_details and list_itsm_tickets, giving you a count of his active tickets across the whole system.

Planning infrastructure upgrades.

The team needs to know what problems might arise from an upcoming software deployment. They ask their agent; it checks list_it_problems and reviews list_change_requests, providing a risk summary before any changes are approved.

Patterns to Avoid

Treating the MCP like a simple search bar

✗ AVOID

Telling your agent to 'tell me about my computer' and expecting it to magically know which asset you mean without context.

✓ INSTEAD

Be specific. To find hardware info, use get_asset_details with an ID or name. If you need user info, start by calling list_requesters first.

Overloading the agent with multiple unrelated requests

✗ AVOID

Asking the agent to 'list users, then check ticket status, and also list assets' all in one massive prompt.

✓ INSTEAD

Break it down. Ask for one thing at a time: First, call list_requesters. Then, based on that output, ask the next question using get_ticket_details.

Assuming current data is always up-to-date

✗ AVOID

Relying solely on the initial response from list_itsm_tickets without checking for status updates.

✓ INSTEAD

Always follow up with a specific check, like calling get_ticket_details and specifying that you need the most recent update time stamp.

The Right Fit

Use this MCP if your day-to-day work involves managing multiple, interconnected systems—specifically IT ticketing (incident management), hardware/software tracking (CMDB), and user data. If answering a question requires checking both an asset record AND an open ticket count, you need this connection. Don't use it if your primary goal is pure HR functions or general accounting; those require different integrations. However, if you only need to create tickets but never check assets, listing out all the tools might be overkill; focus on the ticketing-specific APIs.

The key differentiator here is combining *service desk operations* with *infrastructure data*. If your process flow involves 'User -> Asset -> Problem -> Ticket', this MCP connects every step.

The Dashboard Fatigue

Today, answering a simple question about an employee's laptop—say, finding out when it was last serviced or who currently uses it—is a nightmare. You open the ticketing system to check for related incidents. Then you have to jump over to the separate CMDB portal and manually search by user ID. Finally, if you need the person's department name, you open the HR directory. It's three different logins, five clicks minimum, and half your time is just context-switching.

With this MCP, that sequence of actions vanishes. You tell your agent exactly what you need—for example, 'What are the details on Asset ID X assigned to User Y?' The agent pulls the asset metadata, cross-references the user profile, and gives you a single, coherent answer without you ever leaving your chat window.

Freshservice MCP for Service Desk Operations

You no longer have to manually list tickets via list_itsm_tickets and then use get_ticket_details just to confirm a status. You can ask your agent, 'Are there any high-priority changes needed on the core network?' The tool automatically checks both list_change_requests and list_it_problems simultaneously.

The flow is instant conversation. Instead of running reports or manually updating records across multiple systems, you speak to your agent, and it executes the necessary actions—whether that's creating a new ticket with create_itsm_ticket or listing all users via list_requesters.

Freshservice MCP - 12 Tools

These tools let you automate every core function of IT Service Management, from opening incidents to reviewing asset inventories.

#	TOOL	DESCRIPTION
01	`create_itsm_ticket`	Opens a brand new support ticket in the system with minimal input from you.
02	`check_itsm_status`	Verifies if the helpdesk is currently operational and accepting tickets.
03	`get_asset_details`	Retrieves specific metadata about an IT asset, like its purchase date or assigned user.
04	`get_requester_details`	Pulls the full profile and contact details for a named end-user.
05	`get_ticket_details`	Shows comprehensive metadata, such as priority and status, for a specific ticket ID.
06	`list_support_agents`	Lists all IT staff members who are authorized to work on support tickets.
07	`list_it_assets`	Provides a list of all tracked hardware and software assets in the CMDB.
08	`list_change_requests`	Retrieves records detailing planned changes to the IT infrastructure.
09	`list_it_problems`	Lists known problems or recurring technical issues that need investigation.
10	`list_it_releases`	Accesses records detailing recent software deployments and releases.
11	`list_requesters`	Provides a comprehensive directory listing of all end-users in the company.
12	`list_itsm_tickets`	Generates a list view of all open service desk tickets according to filters you provide.

See It in Action

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

U List all active IT tickets in my service desk.

Retrieving tickets... I found 5 active incidents, including 'VPN Access Failure' (Priority: High) and 'Software Request' (Priority: Low). Would you like the full details for the VPN issue?

U Show me the details for requester 'Jane Smith'.

Searching requesters... I found Jane Smith (jane@example.com). She is in the 'Engineering' department and has 2 open IT tickets associated with her profile.

U Create a new IT ticket: 'Printer Not Working' from 'johndoe@email.com'.

Ticket created! I have successfully opened incident ID 98765 regarding 'Printer Not Working' for johndoe@email.com. It is currently in the 'Open' status.

Frequently Asked Questions

01 **How do I use Freshservice MCP to open a ticket?**

You simply tell your agent you need to file an incident. The tool handles the rest using create_itsm_ticket, requiring only key details like the subject and reporter's email address.

02 **Can Freshservice MCP help me find out who a user is?**

Yes, you use get_requester_details to pull detailed profiles for any employee. This tool gives you their department and contact information instantly.

03 **Does Freshservice MCP track hardware assets?**

Absolutely. You can list all IT assets using list_it_assets, or use get_asset_details to pull specific metadata on a single piece of equipment.

04 **Is it possible to check the current status of my service desk with Freshservice MCP?**

Yes. You can run check_itsm_status at any time to verify if your helpdesk is accepting tickets and functioning normally.

05 **How do I use Freshservice MCP to find old incidents?**

You start by running list_support_agents, which shows who can handle the ticket. Then you refine your search using list_itsm_tickets to narrow down historical records.

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 `"freshservice": { "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

Freshservice 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 Freshservice. 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	Freshservice MCP
Server ID	`019d75a0-b4d1-71eb-beb6-6bd86d65d0a1`
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/freshservice.