MCP SERVER | NO CODE | CLOUD HOSTED

NYC Emergency Response MCP

Get direct access to FDNY and EMS dispatch records.

NYC Emergency Response MCP provides your AI client with direct access to New York City emergency records. You can pull EMS and fire dispatches, locate FDNY firehouses and alarm boxes, analyze historical response times dating back to 2009, and access the FDNY line-of-duty deaths memorial. It works with any MCP-compatible client like Claude, Cursor, or Windsurf.

A+ Quality Score 98.33/100

new-york | nyc | fdny | ems | fire | dispatch

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · SEPTEMBER 2026
vinkius.com/en/ai-agent-connect/nyc-emergency-response-fdny-ems

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.

NYC Emergency Response (FDNY & EMS) MCP

7 tools available
Cloud-hosted on Vinkius

You can now query NYC emergency response data using your preferred AI agent. This MCP pulls directly from NYC Open Data to give you granular details on fire and EMS activity across the five boroughs. If you need to analyze how quickly ambulances are responding to calls in Manhattan versus Brooklyn, you can pull monthly averages or specific incident dispatches. You can also map out the physical infrastructure of the FDNY by locating firehouses and in-service alarm boxes. For researchers or journalists looking at historical trends, the data includes response times going back to 2009 and a memorial list of line-of-duty deaths. Because Vinkius hosts this MCP, you don't have to manage API keys or complex data connections yourself. Just connect your client and start asking questions about NYC's emergency response patterns.

Core Capabilities

01 — Dispatch Analysis

Your agent can query specific EMS and fire incident details including severity and response times.

02 — Infrastructure Mapping

The AI can locate physical FDNY assets like firehouses and alarm boxes using coordinates and addresses.

03 — Historical Trends

Your client can pull monthly response time averages dating back to 2009 for long-term analysis.

04 — Memorial Access

The agent can retrieve specific records from the FDNY line-of-duty deaths memorial list.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/en/ai-agent-connect/nyc-emergency-response-fdny-ems** — connect your AI agent in three steps.

01 Connect your MCP-compatible client to Vinkius.

02 The MCP becomes available as a set of tools in your AI interface.

03 Ask your agent a question about NYC emergency data.

04 The agent calls the appropriate tool to fetch data from NYC Open Data.

05 Your agent processes the raw data and gives you a direct answer.

Connecting to this data is a single-step process through the Vinkius platform.

Built For

This MCP is built for professionals who need to query NYC public safety data without writing custom scrapers or managing API connections.

Emergency Services Researchers

They use the MCP to pull historical response time data for academic or policy studies.

Journalists

They use the tool to investigate local emergency response patterns or incident trends.

Urban Safety Planners

They use dispatch and station location data to inform city safety and resource allocation.

What Changes When You Connect

01 No API keys required to access NYC Open Data.

02 Vinkius handles all hosting and maintenance of the connection.

03 Direct access to historical data back to 2009.

04 Works with any MCP-compatible client you already use.

Real-World Applications

Response Time Audits

Compare average response times across different boroughs to identify service gaps.

Resource Mapping

Identify the proximity of firehouses and alarm boxes to specific neighborhoods or high-risk zones.

Incident Investigation

Look up specific EMS or fire dispatches to understand the nature of recent emergency calls.

Public Safety Reporting

Gather data on fire company incidents to write reports on city-wide emergency trends.

Patterns to Avoid

Requesting massive historical ranges

AVOID

Asking your agent to pull every single EMS dispatch record since 2009 in one go.

✓ INSTEAD

Use specific date windows or filters. The MCP applies automatic date windows to prevent timeouts, so keep your queries focused on specific months or incident types.

Searching for real-time incidents

AVOID

Trying to use this MCP to track a fire that is happening right this second.

✓ INSTEAD

Remember that dispatch data has a lag of a few months. Use `search_fire_dispatches` for recent historical analysis rather than live emergency tracking.

Vague location queries

AVOID

Asking for all firehouses in New York City without any specific parameters.

✓ INSTEAD

Narrow your search by using the borough filter in `list_fdny_firehouses` to get more relevant results.

The Right Fit

Connect this MCP if you are a researcher, journalist, or urban planner needing granular NYC emergency datasets. Do not connect it if you require real-time, live-second emergency dispatch feeds, as the data carries a multi-month lag.

7 Specialized Tools for NYC Emergency Data Access

This MCP provides direct access to FDNY and EMS dispatch records, response metrics, and station locations.

#	TOOL	DESCRIPTION
01	`get_fdny_response_times`	This tool retrieves average FDNY response times for a specific month and borough. It provides historical data dating back to 2009.
02	`list_alarm_boxes`	Use this to find the locations of in-service FDNY alarm boxes. You can filter the results by borough, zip code, or box type.
03	`list_fdny_duty_deaths`	This tool accesses the FDNY memorial dataset. You can filter the list of line-of-duty deaths by specific units or ranks.
04	`list_fdny_firehouses`	This tool provides the locations of FDNY firehouses. Results can be filtered by borough and include addresses and coordinates.
05	`search_ems_dispatches`	This tool searches through FDNY EMS incident dispatches. It includes call types, severity, and response times for a specific date window.
06	`search_fire_company_incidents`	Use this to find specific incidents handled by FDNY fire companies. You can filter by borough, fire box, or incident type.
07	`search_fire_dispatches`	This tool pulls FDNY fire incident dispatches. It provides classification, alarm levels, and response times for recent incidents.

See It in Action

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

U How many fire dispatches happened in the Bronx in the last 30 days?

The search_fire_dispatches tool returns the number of incidents in the Bronx with their classification and response times.

U What was the average FDNY response time in June 2025?

The get_fdny_response_times tool provides the average response time in seconds and incident counts for June 2025.

U List the FDNY firehouses in Staten Island.

The list_fdny_firehouses tool returns all stations in Staten Island, including their addresses, postcodes, and coordinates.

Frequently Asked Questions

01 **Do I need an API key for NYC Open Data?**

No. This MCP provides keyless access to the data through Vinkius.

02 **How recent is the dispatch data?**

There is a data lag of a few months, so the most recent months may appear empty.

03 **How far back does the response time data go?**

You can access historical monthly response time averages dating back to 2009.

04 **Which AI clients can I use with this MCP?**

You can use any MCP-compatible client, including Claude, Cursor, Windsurf, and VS Code.

05 **Can I filter EMS dispatches by borough?**

Yes, you can filter by borough, but the borough names must be uppercase in the dispatch dataset.

06 **Do I need an API key?**

No. NYC Open Data serves all public datasets anonymously over its SODA API (data.cityofnewyork.us). This MCP defines no credentials and needs nothing configured.

07 **How current are the dispatch datasets?**

Dispatch rows lag a few months behind real time (the newest rows stop around mid-2026), so both dispatch search tools apply a 90-day window by default. Monthly response-time statistics are historical, back to 2009, and are stored as "YYYY/MM" — the year_month parameter accepts "YYYY-MM" and normalizes it.

08 **Which borough values should I use?**

Dispatch data stores boroughs in UPPERCASE — "BRONX", "BROOKLYN", "MANHATTAN", "QUEENS", "RICHMOND / STATEN ISLAND" (Staten Island), UNKNOWN. Location lists (firehouses, alarm boxes, fire-company incidents) use title case — "Staten Island", "Manhattan". Each tool states its case in its description.

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 `"nyc-emergency-response-fdny-ems": { "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

NYC Emergency Response (FDNY & EMS) 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 NYC Emergency Response (FDNY & EMS). 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	September 2026
MCP Server	NYC Emergency Response (FDNY & EMS) MCP
Server ID	`01a0dfcd-55c9-7380-8c61-94da9b79c65e`
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/en/ai-agent-connect/nyc-emergency-response-fdny-ems.