MCP SERVER | NO CODE | CLOUD HOSTED

Datadog AI LLM Observability MCP for AI Agents

Monitor token usage and track model performance metrics in production systems

Datadog AI (LLM Observability) MCP allows you to monitor, audit, and track performance metrics for your LLMs in real-time. It lets your agent pull high-precision data on token usage, latency spikes, prompt content, and overall infrastructure health directly from your existing Datadog setup.

A+ Quality Score 100/100

llm-observability | token-usage | prompt-monitoring | ai-performance | telemetry | model-auditing

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/datadog-ai-llm-observability

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.

Datadog AI (LLM Observability) MCP

10 tools available
Cloud-hosted on Vinkius

Running models is complex; tracking their cost and performance shouldn't be. This MCP connects your AI client to your Datadog account so you can manage LLM observability through natural conversation. Instead of hopping between dashboards and logs, your agent handles the deep dive. You can query metrics for specific things like token counts or latency timeseries, pull full prompt logs, and even check active outages that might be blocking multi-agent workflows. It also lets you view widgets graphing global AI expenses across providers like OpenAI and Anthropic.

When you connect this MCP via Vinkius, your agent gets immediate visibility into every part of your model stack—from simple usage tracking to complex incident reporting. You'll know exactly when a dynamic LLM model was switched out or if performance is starting to drop below established thresholds.

Core Capabilities

01 — Querying Token and Latency Metrics

Find the average token usage, peak consumption times, and overall latency for your models over specific periods.

02 — Auditing Prompt Content and Model Spans

Retrieve detailed records of literal prompts and response traces, helping you debug exactly what inputs caused performance issues.

03 — Checking for Active Service Outages

Monitor your infrastructure to detect real-time service disruptions or active outages blocking agent workflows.

04 — Creating Performance Alerts

Set up monitors that alert you when AI responses drop below expected performance levels or hit resource limits.

05 — Analyzing Global AI Infrastructure Spending

Enumerate widgets that graph total global spending and usage across different LLM providers, aiding budget planning.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/datadog-ai-llm-observability** — connect your AI agent in three steps.

01 Subscribe to this MCP in Vinkius and provide your Datadog API Key, APP Key, and Site details.

02 Your AI client authenticates using these credentials, granting the necessary read permissions for observability data.

03 You simply ask your agent a question—like 'What was my token usage last quarter?'—and it fetches the precise metrics from your infrastructure.

The bottom line is that you get direct, natural language access to highly technical performance and financial logs without ever leaving your chat window.

Built For

This MCP is for the MLOps Engineer who needs real-time visibility into model costs. It's for the SRE tired of manually checking dashboards when an AI service hiccups, and it’s for the FinOps analyst needing precise proof of LLM spending.

MLOps Engineer

Audits prompt logs and traces to track model performance across different versions and identifies specific resource bottlenecks.

SRE (Site Reliability Engineer)

Sets up monitors for AI services, tracks service disruptions, and verifies agentic workflows are functioning correctly during an incident.

FinOps Analyst

Analyzes dashboards that graph global AI infrastructure expenses and usage patterns to optimize spending across providers.

What Changes When You Connect

01 Track actual resource consumption by querying specific metrics, like average tokens per request or latency spikes, using `query_metrics`.

02 Never miss an outage. Use `list_incidents` to get real-time updates on service disruptions that could halt your agentic workflows.

03 Manage performance automatically by calling `create_monitor`, setting alerts for when model responses fall below acceptable thresholds.

04 Keep a clean audit trail of every interaction. Utilize `search_llm_spans` to retrieve the exact prompt and response payload contents needed for debugging.

05 Control your costs proactively. You can view global spending patterns by using `list_dashboards`, giving you financial oversight across all model providers.

Real-World Applications

Debugging a sudden spike in costs

A developer noticed their monthly LLM bill was spiking. They asked their agent to check the logs, which used `search_llm_spans` to retrieve specific payloads and pinpointed that a single unoptimized prompt loop was causing excessive token usage.

Verifying model stability after an update

The MLOps team just rolled out Model v2. They used `list_ai_monitors` to check if all their existing performance monitors were still tracking correctly and confirmed that the new version maintained low latency metrics.

Diagnosing agent failure during peak hours

When an automated workflow failed, the SRE used `list_incidents` to check for active service disruptions. The report showed a temporary gateway authentication failure that was blocking multi-agent orchestration.

Optimizing cloud spending across multiple services

The FinOps team needed an overall view of AI spend. They used the MCP to enumerate global expenses, allowing them to compare usage patterns between OpenAI and Anthropic in one place.

Patterns to Avoid

Checking logs manually

✗ AVOID

Having to jump into Datadog's dashboard, filter by 'LLM,' then search for specific time ranges just to find out why latency spiked yesterday afternoon.

✓ INSTEAD

Just ask your agent. Use `query_metrics` and specify the exact timeframe you care about. The MCP handles all the filtering and data retrieval in one go.

Missing a service outage

✗ AVOID

Assuming that because the application *seems* fine, nothing is wrong. You might miss a subtle background failure or an active incident blocking core functionality.

✓ INSTEAD

Always run `list_incidents` first. It checks for known and active outages before you start debugging specific model issues.

Guessing the root cause of high tokens

✗ AVOID

Seeing a massive token count but not knowing *which* prompt or which user interaction caused it, leading to wasted time and inaccurate cost reports.

✓ INSTEAD

Run `search_llm_spans`. This gives you the full, literal payload context—the exact input and output that drove the high usage.

The Right Fit

Use this MCP if your primary pain point is visibility into LLM performance, cost, or infrastructure health. If you need to know *why* a model was slow or expensive, this tool's access to detailed metrics and spans is critical. Don't use it if all you need is basic usage counting; simple billing APIs might suffice. However, if your issue involves diagnosing the root cause—like linking a spike in token counts back to a specific prompt structure—you absolutely need `search_llm_spans`. If you are only interested in general system status and not LLM-specific metrics, another type of infrastructure monitoring tool will work better.

Datadog AI LLM Observability: Auditing Prompt Logs for Model Debugging

Manually debugging an LLM pipeline is a nightmare. You spend hours switching between the application logs, the metrics dashboard, and the billing console. You try to find out why latency spiked on Tuesday morning or which prompt caused that massive token burst—it's a painful copy-paste cycle across multiple tabs.

With this MCP, you ask your agent, 'Show me all prompts run by Agent X between 9 AM and 10 AM.' The tool uses `search_llm_spans` to pull the full JSON payload content directly. You get the exact prompt logic, the response trace, and the associated metrics in a single chat output.

Datadog AI LLM Observability: Tracking Infrastructure Costs and Alerts

Before connecting this MCP, figuring out your total cost meant running separate reports for OpenAI, Anthropic, and internal compute. You had to piece together usage patterns from various billing systems, making it impossible to see the global picture.

Now, you can ask the agent, 'What is our projected spend next month if we increase volume?' The MCP pulls dashboard insights, giving you a unified graph of AI expenses across providers instantly. It shifts cost management from reactive auditing to proactive planning.

Datadog AI LLM Observability: 10 Tools for Model Performance Auditing

These tools let your agent perform deep checks on performance metrics, track service incidents, list available dashboards, and audit detailed usage spans.

#	TOOL	DESCRIPTION
01	create_event	Inspects deep internal arrays related to plan math calculations for debugging purposes.
02	create_monitor	Creates explicit validation checks, allowing you to monitor specific metrics or thresholds automatically.
03	list_dashboards	Retrieves a list of structured rules attached to billing accounts for monitoring purposes.
04	list_events	Identifies precise active arrays spanning native gateway authentication records.
05	list_incidents	Dispatches an automated validation check to route explicit historical service outage data.
06	search_llm_spans	Searches for detailed JSON payload contents, providing hard customer usage bindings and context.
07	list_ai_monitors	Retrieves explicit cloud logging information that traces resource limits associated with AI models.
08	query_metrics	Queries core LLM observability metrics, such as token count and latency, from the platform.
09	submit_series	Performs structural extraction of properties that drive active account logic changes.
10	list_service_accounts	Identifies precise active arrays spanning native hold parsing records for service access management.

See It in Action

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

U Can you show me the token usage and latency for our main chatbot over the last four hours?

LLM Performance Metrics (Last 4 Hours)

METRIC	AVERAGE VALUE
Tokens/Request	185 tokens
Latency	920 ms
Max Usage Spike	3.1k tokens (1:30 PM)

The average latency is stable, but usage spiked near 1:30 PM. This suggests a high-complexity query ran during that time.

U I suspect an outage happened yesterday; list any active service disruptions.

Incident Report

✅ **Status:** Multiple incidents detected.
- **Service:** Primary Gateway Auth
- **Impact:** Intermittent failure (10/26, 4:00 PM - 4:30 PM EST)
- **Details:** Authentication errors blocked multi-agent orchestration. The service was restored automatically.

U List all the current AI performance monitors we have set up.

Active Monitoring Status

You currently have 5 monitors running:

- [LLM-Latency-High]: **ALERT**. Average latency exceeded threshold (1200ms).
- [Token-Quota-Reached]: OK.
- [Model-Drift]: OK. Needs review.
- [GPU-Utilization]: OK.
- [ErrorRate-Threshold]: OK.

Frequently Asked Questions

01 **How does the Datadog AI LLM Observability MCP help me track costs?**

It provides a unified view of your spending. Instead of checking separate billing portals for every provider, you can ask the agent to graph global expenses and see exactly which models are driving your highest costs.

02 **I need to debug a failed LLM workflow; what should I use with this MCP?**

Use the tool that searches for LLM spans. It lets you pull the full prompt payload and response traces, showing you exactly which input caused the failure or poor output.

03 **Can this MCP tell me if my AI services are currently down?**

Yes. By listing incidents, your agent checks for active outages and service disruptions across your entire infrastructure, ensuring that a simple background failure won't break your workflow.

04 **How do I set up alerts for poor model performance using the Datadog AI LLM Observability MCP?**

You can use the capability to create monitors. You tell the agent what threshold you care about, and it sets up an alert that notifies you when the latency or token usage gets too high.

05 **Is this Datadog AI LLM Observability MCP better than just checking raw logs?**

It's much better. Instead of drowning in raw, unstructured data, the MCP interprets those logs and presents you with actionable metrics—like average usage or specific failure points—in plain language.

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 `"datadog-ai-llm-observability": { "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

Datadog AI (LLM Observability) 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 Datadog AI (LLM Observability). 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	Datadog AI (LLM Observability) MCP
Server ID	`019d7581-a5af-72b6-a2cf-684e1f80d513`
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/datadog-ai-llm-observability.