MCP SERVER | NO CODE | CLOUD HOSTED

Netdata MCP

Get Real-Time Infra Metrics and Alerts Instantly

Netdata MCP connects your entire infrastructure monitoring suite to any AI client. Instantly pull real-time performance metrics, check current system alerts, and get deep operational data for specific nodes or across entire cloud spaces. Your agent reads the raw numbers so you don't have to click through dozens of dashboards.

F Quality Score 43.65/100

real-time-monitoring | infrastructure-observability | system-metrics | performance-analysis | alert-management

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

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.

Netdata MCP

10 tools available
Cloud-hosted on Vinkius

Your AI client can connect directly to Netdata to give you instant visibility into your infrastructure's performance. Instead of jumping between monitoring tabs, you ask a question—like 'Why is CPU spiking on Node Alpha?'—and get an immediate answer backed by real data. You can fetch granular metrics for specific components like RAM or network throughput using one command. Need to know if something is broken? Your agent checks active alarms across your local machines or even monitors critical issues space-wide. When you're ready to analyze the raw numbers, you retrieve all collected metrics in a format that external tools love. This makes troubleshooting faster and less painful. Through Vinkius, you connect this powerful MCP with any compatible client, giving your agent 24/7 System Administrator capabilities right where you work.

Core Capabilities

01 — Analyze system performance data

Fetch detailed metric readings—like CPU load or disk usage—for a specific component on a machine.

02 — Manage current alerts

Check for active warnings and alarms, either on a single local agent or across an entire cloud environment.

03 — View infrastructure inventory

List all connected spaces, rooms, and individual nodes to understand your full monitoring scope.

04 — Export raw metrics data

Collect every available metric across the monitored nodes into a single dataset for external processing.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Netdata Cloud Token or Agent URL.

02 Your AI client accesses the connection details, giving it visibility into all connected nodes and spaces.

03 You prompt your agent with a question (e.g., 'What are the current disk alerts?') and receive specific, actionable data.

The bottom line is you get immediate operational insights without ever leaving your IDE or terminal.

Built For

This MCP is for ops engineers who spend too much time clicking through dashboards at 2 AM. It's essential for SREs and System Administrators managing large, complex environments that require instant data correlation.

DevOps Engineer

Correlate system alerts with recent deployments by asking your agent to check current alarms against deployment logs.

SRE (Site Reliability Engineer)

Speed up incident response by having the MCP automatically gather specific chart data or list critical space-wide issues.

System Administrator

Manage multi-node environments efficiently by listing all connected spaces and rooms via simple conversation prompts.

What Changes When You Connect

01 Instant bottleneck diagnosis: Instead of guessing, ask your agent to fetch metric data from a specific chart using `get_chart_data` to pinpoint exactly where performance is dipping.

02 Centralized alert visibility: Check for local machine issues with `get_alarms`, or get a high-level view of critical alerts across the whole space using `list_space_alerts`.

03 Full inventory mapping: Never lose track of your assets. Use `list_spaces` and `list_rooms` to map out every node connected through your infrastructure.

04 Historical data analysis: Need to feed metrics into a custom tool? Run `get_all_metrics` to pull all raw performance numbers for deep external processing.

05 Quick health checks: Before starting an investigation, run `get_agent_info` to confirm the version and basic status of the agent itself.

Real-World Applications

Investigating a sudden network slowdown

The user asks their agent about recent network issues. The agent checks `list_space_nodes` to identify all relevant machines, then uses `get_chart_data` on the network metrics for those nodes. It returns specific data confirming which machine is spiking and why.

Performing a routine system audit

The user needs a full picture of the environment. The agent first calls `list_spaces` to see all environments, then uses `list_rooms` to drill down into each one, building a complete map of connectivity.

Responding to an urgent production alarm

A critical alert hits. The user prompts the agent for active alarms using `get_alarms`. The agent immediately returns which specific component is failing and whether other system health checks are clear, saving minutes of manual investigation.

Preparing data for a quarterly report

The team needs historical performance metrics. Instead of exporting manually from dashboards, the user runs `get_all_metrics`, getting all raw data in one go that can be fed directly into reporting tools.

Patterns to Avoid

Manually checking every dashboard

✗ AVOID

A sysadmin has to open 15 different tabs, click through menus, and copy/paste metrics from CPU graphs, RAM graphs, and disk status pages one by one.

✓ INSTEAD

Ask your agent to run `list_space_nodes` first. Then ask it to fetch the specific performance data using `get_chart_data`. It handles the navigation for you.

Ignoring high-level alerts

✗ AVOID

The system is failing due to a critical, space-wide issue, but the engineer only checks the local machine's dashboard and misses the root cause.

✓ INSTEAD

Always run `list_space_alerts` first. This function aggregates issues across all connected nodes and gives you the full picture immediately.

Confusing metrics with inventory

✗ AVOID

The user asks 'What is wrong?' but doesn't know if they need a metric check or an asset list, leading to vague results.

✓ INSTEAD

If you suspect poor performance, use `get_chart_data`. If you just need to map your infrastructure, use the combination of `list_spaces`, `list_rooms`, and `list_space_nodes`.

The Right Fit

Use this MCP if your job involves high-volume incident response or managing environments with dozens of nodes. You need a single point of access that can query metrics, check alerts, and map out the topology without you having to click through UIs. Don't use it if you only need to view one specific metric (like just CPU). For simple checks, just viewing charts might be enough. But if you need correlation—for instance, 'Show me all nodes in Space X that have active alerts AND whose RAM usage was over 90% last hour'—then this MCP is your only option. It combines the functions of `get_alarms`, `list_space_nodes`, and `get_chart_data` into one conversational flow.

The Dashboard Overload Problem

Today, finding out why a service is slow means logging into dashboard A to check CPU, then switching to dashboard B to see disk I/O. If you need to know about alerts, you open dashboard C. You end up copy-pasting data across spreadsheets just to build a picture of the problem.

With this MCP, all those separate views disappear. You simply ask your agent what's wrong with the service. It correlates metrics from CPU and disk checks, reports on any active alarms via `get_alarms`, and gives you one single answer right in your chat window.

Get Actionable Insights with Netdata MCP

You stop manually checking node versions using the agent's ability to run `get_agent_info`. You don't have to track down which spaces are connected; listing them all via `list_spaces` is instant.

The difference is that you get direct, real-time answers. Your AI client acts as a system expert who has already checked every tab for you.

Netdata MCP: 10 Tools for Monitoring

Use these tools in conversation with your AI agent to query everything from specific charts to entire cloud space alert statuses.

#	TOOL	DESCRIPTION
01	`get_alarms`	Retrieves the current status of every configured alarm on a machine.
02	`get_all_metrics`	Gathers all available performance metrics into one bulk dataset for scraping or analysis.
03	`list_charts`	Provides a list of every metric chart currently tracked on the connected node.
04	`get_chart_data`	Fetches specific time-series data points from any requested performance chart.
05	`get_agent_info`	Returns detailed information, including the version and host details, for the Netdata Agent.
06	`list_room_nodes`	Lists all individual nodes belonging to a specific monitored room.
07	`list_rooms`	Provides a list of all rooms within a defined monitoring space.
08	`list_space_alerts`	Aggregates and reports on critical issues across the entire monitored cloud space.
09	`list_space_nodes`	Lists all machines connected under a specific monitoring space.
10	`list_spaces`	Retrieves a comprehensive list of all available Netdata Cloud spaces.

See It in Action

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

U Get the current information and version of the Netdata agent.

I've retrieved the agent info. The host is 'prod-server-01' running Netdata version v1.44.1 with 128 enabled collectors.

U List all available charts on this node so I can see what metrics are being collected.

I've compiled the list of charts. You have access to metrics for 'system.cpu', 'system.ram', 'net.eth0', and 45 other dimensions. Which one would you like to analyze?

U Are there any active alarms or warnings on the local agent right now?

Checking alarms... I found one warning for 'disk_backlog' on /dev/sda. All other system health checks are currently clear.

Frequently Asked Questions

01 **How do I check all nodes connected to my environment using Netdata MCP?**

You can list all spaces first using `list_spaces`. Then, drill down into the rooms and finally use `list_space_nodes` to get a complete inventory of every machine monitored.

02 **Can I check for active alarms with Netdata MCP?**

Yes. You can run `get_alarms` to see local warnings, or you can use `list_space_alerts` if you need a summary of critical issues across the entire cloud space.

03 **What data do I get when I use get_chart_data with Netdata MCP?**

You get specific, time-series metric readings for any chart type, like CPU or RAM. This allows you to diagnose performance bottlenecks precisely rather than just seeing a general graph.

04 **Does Netdata MCP help me analyze historical data?**

Yes, you can use `get_all_metrics` to retrieve all collected metrics in a format designed for external analysis tools, making history accessible.

05 **Is Netdata MCP just for Linux servers?**

The MCP is designed to work with your configured Netdata Agent. It accesses the data available through that specific agent connection.

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 `"netdata": { "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

Netdata 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 Netdata. 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	Netdata MCP
Server ID	`019e38c7-3a25-7353-8181-983dc35217b4`
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/netdata.