MCP SERVER | NO CODE | CLOUD HOSTED

Sentry Alternative MCP

Monitor errors and audit releases via conversation.

Sentry Alternative MCP connects your AI agent directly to application error data, letting you monitor system health and debug issues without leaving your chat window. Query full stacktraces, check deployment versions, and audit alert rules from any MCP-compatible client.

A+ Quality Score 100/100

error-tracking | performance-monitoring | application-health | debugging | alerting | issue-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/sentry-alternative

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.

Sentry MCP

15 tools available
Cloud-hosted on Vinkius

Your app is breaking, and the dashboard is a nightmare of clicks. This MCP changes that by giving your AI agent real-time access to all your application errors. Instead of switching between tabs to find out what went wrong, you just ask your agent to look into it. It pulls raw error events with full stacktraces and context—like HTTP headers and user data—so you can debug the root cause immediately.

It lets you track which version of code caused a problem by listing all deployments. You can also get a complete picture of your team's notification system, reviewing every configured alert rule to make sure no critical issue gets missed. If you use Vinkius, you connect this MCP alongside thousands of others, giving your agent one central place to manage observability and debugging.

It’s like having an on-call engineer sitting next to you, ready to pull up the data whenever a problem pops off.

Core Capabilities

01 — Inspect Raw Error Events

Retrieve complete details for specific errors, including stacktraces and user context.

02 — Search and Group Issues

Filter across all projects or the entire organization to find unresolved issues based on status or priority.

03 — Track Deployments and Releases

List every version deployed, allowing you to link specific bugs back to their source code release.

04 — Audit Alert Rules

Review all active notification rules (for Slack, email, PagerDuty) across your organization's projects.

05 — Update Issue Statuses

Change the status of a reported issue or assign it to another team member directly through conversation.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your Sentry Internal Integration Token.

02 Your AI client verifies the connection, giving you immediate access to organizational settings.

03 You ask a natural language question—like 'Show me all unresolved payment errors from last week'—and receive structured data directly.

The bottom line is that your agent handles the complex API calls and delivers clean, actionable information right where you're working.

Built For

This MCP is for developers, on-call engineers, and engineering managers who get frustrated with context switching. If you spend your nights jumping between dashboards just to find out why a bug appeared after the last deployment, this is for you.

On-Call Engineer

Triaging new errors instantly by asking the agent to list recent events or check if an issue was already resolved.

Software Developer

Debugging regressions by inspecting event breadcrumbs and checking which specific release introduced a problem.

Engineering Manager

Auditing overall system health by viewing open issue counts across multiple projects or reviewing alert rules.

What Changes When You Connect

01 Stop manually opening the dashboard. Instead, ask your agent to list recent events or search issues by priority status, getting instant answers without context switching.

02 Pinpoint regressions fast. By calling get_release and viewing deployment metadata, you can correlate a specific bug directly back to the version that introduced it.

03 Review notification paths completely. Use list_alert_rules to audit every single rule—from Slack messages to PagerDuty triggers—to ensure your team gets timely warnings.

04 Deep dive debugging is easier. The agent pulls full event details, giving you complete stacktraces and user data via get_event, far beyond what a simple dashboard view shows.

05 Manage the cleanup process. You can update an issue's status or assign it to someone else using update_issue, keeping your project records accurate without needing to click into the UI.

Real-World Applications

Debugging a production failure

A developer notices high error rates in the payments service. They ask their agent to list recent events for that project, find the specific `ZeroDivisionError`, and use get_event to pull the full stacktrace, identifying the line of code responsible.

Auditing team coverage

An EM needs to know if the on-call rotation is covered for all critical services. They ask their agent to list projects and then use list_alert_rules to confirm that every core service has a corresponding PagerDuty trigger.

Investigating stale issues

A team member suspects an old bug is still open. Instead of browsing manually, they ask the agent to search issues using specific query syntax and get_issue details on the top result to verify its current status.

Checking deployment impact

The product team needs to know if a new feature caused an issue. They prompt their agent with the date range, which triggers list_releases and then allows them to search issues scoped only to that specific version.

Patterns to Avoid

Manually searching error logs

✗ AVOID

Opening Sentry, navigating to the project selector, setting date ranges, and using the built-in filters just to find a handful of similar errors.

✓ INSTEAD

Tell your agent to search issues directly. Use list_issues or search_issues with parameters like 'is:unresolved priority:50' for instant results.

Assuming full context is available

✗ AVOID

Looking at a simple error summary on the dashboard and guessing the root cause, only to find out later that critical breadcrumbs were missing.

✓ INSTEAD

Always ask your agent to get_event. This pulls all raw data, including complete stacktraces, HTTP context, and user data.

Ignoring deployment history

✗ AVOID

Fixing a bug but not knowing if the fix was actually introduced by the latest release or an older one.

✓ INSTEAD

Run list_releases to track every version. Then, use get_release on the suspicious version to correlate issues precisely.

The Right Fit

Use this MCP when your primary goal is deep observability and debugging. If you need to know *why* something broke, or *when* it broke, this tool suite gives you everything: event data (get_event), historical context (list_releases), and systemic visibility (list_alert_rules). Don't use it if you simply need a quick status check on user accounts; other tools are better suited for that. However, if your workflow involves anything related to code quality, deployments, or error triage, this is the standard toolset. It's designed to turn dashboard navigation into conversational querying.

The Overhead of Dashboard Context Switching

Right now, when an alert fires—say, a critical payment processing failure—you're forced into a painful routine. You have to open the Sentry dashboard, navigate to the right project, filter by date, find the specific error type, click through multiple panels just to see the full stacktrace, and then copy the information out. It takes time, and you lose critical context every time you jump between tabs.

With this MCP connected via Vinkius, that process collapses into a single query. You tell your agent what's wrong—for example, 'What happened with user signups yesterday?'—and it pulls the full error events and details straight to your chat window. The outcome is immediate visibility without ever leaving your development workflow.

Actionable Error Data via Sentry Alternative MCP

Instead of manually cross-referencing issues, you can use the agent to list all projects and then run a targeted search using search_issues. You'll check status, priority, and first/last seen timestamps all in one go, eliminating hours of manual filtering.

This MCP lets your agent manage complex tasks like updating an issue or retrieving release details with simple commands. It doesn't just show you data; it executes actions, making debugging faster and more reliable.

Sentry Alternative MCP with 15 Tools

These tools give your agent specific functions: list organizations, search issues, check event details, or update issue statuses. Use them to get full system visibility.

#	TOOL	DESCRIPTION
01	`list_alert_rules`	Retrieves a list of all configured alerts, detailing their conditions and actions (like sending an email or Slack message).
02	`get_auth_info`	Checks if your integration token is active and correctly connected to the Sentry account.
03	`list_events`	Lists recent error events for a specific project, providing key details like timestamps and basic stacktrace snippets.
04	`get_event`	Fetches the full, detailed record for an event ID you obtained from listing recent errors.
05	`get_issue`	Gets comprehensive details about a specific issue using its unique numeric identifier.
06	`get_project`	Retrieves the full configuration and metadata for any specified project within your organization.
07	`get_release`	Pulls all deployment metadata for a specific version string, linking it to the organization.
08	`list_issues`	Lists issues across your organization or scoped to one project; you can filter results by status or priority.
09	`list_organizations`	Retrieves a list of all organizations you belong to, providing their unique slugs for subsequent calls.
10	`list_projects`	Lists every application or service project within your organization, requiring the parent organization slug.
11	`list_releases`	Gets a list of all deployed versions for an organization or a specific project to track history.
12	`search_issues`	Searches issues using Sentry's query syntax, allowing you to narrow results by text, priority, or status across projects.
13	`list_tags`	Displays all available tags used for categorization within a specific organization or project.

#	TOOL	DESCRIPTION
14	`list_teams`	Lists the various teams within your organization, using the organization slug to scope the results.
15	`update_issue`	Changes the status or adds/removes tags from an existing issue using its numeric ID.

See It in Action

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

U Show me all unresolved issues in my backend-api project.

I've scanned your backend-api project and found 4 unresolved issues. The most critical is a `ZeroDivisionError` in payment processing (127 events in the last 24h). Would you like the full stacktrace?

U Which releases have been deployed for my organization in the last month?

I've retrieved 6 releases for your organization. The most recent is `v2.14.3` deployed 2 days ago to the production environment. The oldest in this window is `v2.12.0` from 28 days ago. Would you like event counts per release?

U What alert rules are currently configured for the mobile-app team?

Your mobile-app project has 3 active alert rules: 1) Notify #eng-mobile on Slack when a new issue is created, 2) Send email to oncall@company.com when issue count exceeds 100 in 5 minutes, 3) Create PagerDuty incident for errors with priority 50. Would you like the full configuration details for any rule?

Frequently Asked Questions

01 **How do I check the status of a bug using Sentry Alternative MCP?**

You can use list_issues to see all open bugs across your organization. If you need details on one specific issue, use get_issue with its numeric ID.

02 **Can I find out which code version caused the error using Sentry Alternative MCP?**

Yes, run list_releases to see all deployed versions. You can then reference those versions when searching issues or getting details for a specific project release.

03 **What if I need to know who gets notified when things break with Sentry Alternative MCP?**

You run list_alert_rules. This tool shows you all configured alerts, letting you audit the conditions (e.g., issue count > 100) and the resulting actions (Slack, email).

04 **Is Sentry Alternative MCP good for debugging stacktraces?**

Absolutely. Use get_event to retrieve raw error events that include complete stacktraces, breadcrumbs, and all relevant HTTP context.

05 **Does the Sentry Alternative MCP allow me to change an issue's status?**

Yes, you can use update_issue. This allows your agent to change the status of a bug or assign it to another team member without manual UI clicks.

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 `"sentry-alternative": { "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

Sentry 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 Sentry. 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	Sentry MCP
Server ID	`019d847d-0025-72e5-95e4-efa42cfc6872`
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/sentry-alternative.