MCP SERVER | NO CODE | CLOUD HOSTED

SonarCloud MCP

Check code safety and quality metrics instantly.

SonarCloud MCP lets you bring professional static code analysis directly into your AI agent's conversation. Instead of opening dashboards or running manual checks, your agent queries project bugs, technical debt metrics, and security hotspots instantly. Use it to ensure the code structure is secure, compliant, and ready for production without leaving your editor.

A+ Quality Score 100/100

static-analysis | code-quality | technical-debt | security-hotspots | ci-cd-pipeline | code-review

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.

SonarCloud MCP

9 tools available
Cloud-hosted on Vinkius

Stop switching between your IDE and a separate quality dashboard just to check if merging that pull request is safe. This MCP connects SonarCloud's industry-leading analysis tools straight into your AI agent. You can ask things like, "What's the current code coverage on the payment service?" and get an immediate answer with metrics. It lets you verify project bugs, assess technical debt, and check for security vulnerabilities—all through natural chat. The system ensures that any code written or reviewed by your agent adheres to your organization's strict CI/CD rules. Through Vinkius, this MCP makes SonarCloud's powerful analysis capabilities available wherever your AI client connects.

Core Capabilities

01 — Check Project Health Status

Use `get_quality_gate_status` to instantly check if a project passed all mandatory quality checks.

02 — Find Code Flaws and Vulnerabilities

Search for specific code quality issues using `search_issues` or get deep details on any finding with `get_issue_details`.

03 — Analyze Project Structure

Discover application projects via `search_projects` and map out the internal components of a codebase using `list_project_components`.

04 — Retrieve Key Metrics

Pull specific quality metrics, like code coverage percentages, for any project component using `get_project_measures`.

05 — Review Organization and Users

List all connected organizations with `list_organizations` or search for team members in your directory via `search_users`.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP within Vinkius and introduce your SonarCloud Security Token.

02 Tell your AI client the specific project or code base you want analyzed (e.g., 'Check the coverage for Project X').

03 Your agent runs the necessary checks, returning actionable data points like failure status or vulnerability counts directly in the chat.

The bottom line is that your AI client accesses SonarCloud's entire analysis suite without you needing to log into a single web dashboard.

Built For

This MCP is built for developers and engineers who get frustrated by context switching. It's for the developer who hates clicking through dashboards at 2 am just to confirm if a PR can merge, or the DevSecOps specialist who needs immediate proof of failing quality gates before approving code.

Software Developer

Uses this MCP to ask for quick checks on new components—for example, confirming that local coverage meets minimum standards before pushing a commit.

DevSecOps Engineer

Queries the exact details of failing quality gates or critical vulnerabilities (like hardcoded tokens) right from their terminal context.

Team Lead / Architect

Gathers accurate, historical metrics—such as total lines of code across different projects or overall technical debt status—without opening the main reporting UI.

What Changes When You Connect

01 Stop digging through SonarCloud's UI. You ask your agent about `get_quality_gate_status` and get a clear pass/fail status without switching tabs.

02 Never manually search for bugs again. Use `search_issues` to filter only for CRITICAL or MAJOR issues, giving you an actionable list immediately.

03 Understand the full scope of your codebase by using `list_project_components` to map out every file and directory within a project's hierarchy.

04 Know if your code is safe before merging. You can use `get_project_measures` to pull specific metrics, like coverage percentage, right in your chat window.

05 Manage team visibility easily by listing all connected organizations with `list_organizations` or finding users via `search_users`.

Real-World Applications

Reviewing a PR before Merge

A developer asks their agent, "What's the quality gate status for the API service?" The agent runs `get_quality_gate_status` and reports: 'ERROR. Code coverage dropped to 74% (below mandatory 80%). You need to fix this before merging.' This prevents a broken release cycle.

Finding Hidden Vulnerabilities

A DevSecOps engineer wants to audit the authentication module. They tell their agent, "Find all CRITICAL vulnerabilities in `src/auth/`." The agent uses `search_issues` and immediately flags a hardcoded token found via `get_issue_details`, preventing a security breach.

Assessing Project Scope

A Team Lead needs to know how many components are in the new payment service. They instruct their agent to use `list_project_components` to generate a full, accurate list of all files and directories for review.

Mapping Organizational Access

A manager needs an overview of who can access which code base. They ask the agent to run `list_organizations` followed by `search_users`, generating a clean list of all connected entities and their active users.

Patterns to Avoid

Manual Dashboard Checking

✗ AVOID

A developer has to open the SonarCloud web UI, navigate to the 'Quality Gate' tab, filter by severity, and then manually check coverage metrics in a separate pane.

✓ INSTEAD

Just ask your agent. It runs `get_quality_gate_status` for you. The result is delivered instantly, keeping everything right where you are.

Guessing Vulnerabilities

✗ AVOID

A team member suspects a vulnerability but doesn't know the exact key or path to check in the system.

✓ INSTEAD

Use `search_issues` first. You can narrow down the search by component, then use `get_issue_details` to get full context and remediation steps.

Overloading the Agent

✗ AVOID

Asking the agent a vague question like "Tell me about the code quality." The response is unhelpful because it lacks specific metrics.

✓ INSTEAD

Be specific. Ask, "What are the project measures for code coverage in `api-backend-core`?" This directs the agent to use `get_project_measures` and get real data.

The Right Fit

Use this MCP if your primary concern is code compliance, security vulnerability tracking, or quantitative metrics. If you need to know *if* a PR should merge based on mandatory rules, or if the codebase has hardcoded secrets, this is essential. Don't use it if you just need general suggestions for improving readability or refactoring small chunks of code; those are better handled by your agent's base coding features. This MCP is strictly about high-level, audited quality checks, making tools like `get_quality_gate_status` and `search_issues` non-negotiable parts of your workflow.

The pain of context switching on code review

Today, reviewing a PR means leaving the flow. You write the code in your editor; then you jump to SonarCloud's website, find the project key, click the Quality Gate tab, read a report that says 'Failed,' and then copy-paste findings back into your chat window. It's clicking through five different tabs just to answer one question.

With this MCP, that entire process disappears. You tell your agent what you want checked—say, "Is the payment service ready?" The agent runs all the necessary checks, pulls the metrics, and hands you a clear verdict right in the chat. It's immediate, focused, and keeps you where you belong: writing code.

Get SonarCloud Quality Status with `get_quality_gate_status`

Previously, confirming compliance meant running a full CI/CD pipeline and waiting minutes for the dashboard to update. You'd have to remember which project key and which branch were being tested.

Now, you just ask your agent to check the status using `get_quality_gate_status`. The system handles the complex API calls and reports back instantly: Pass or Fail. That's it.

SonarCloud MCP with 9 Tools

These tools let you programmatically interact with SonarCloud data, allowing your agent to search for issues, list projects, and retrieve specific quality metrics.

#	TOOL	DESCRIPTION
01	`get_issue_details`	Retrieves specific information about a single reported code quality issue.
02	`get_project_measures`	Fetches various quality metrics, like coverage, for an entire project component using its key and metric name.
03	`get_analysis_status`	Pulls the most recent status of a code analysis run for any given project.
04	`get_quality_gate_status`	Checks if a project's quality gate passed or failed, providing an immediate pass/fail signal.
05	`list_project_components`	Lists all files and directories that make up the components within a target project.
06	`list_organizations`	Retrieves a list of organizations connected to your user account.
07	`search_issues`	Searches the codebase for issues, allowing you to filter results by component key or severity level.
08	`search_projects`	Locates and lists available projects within your SonarCloud account using an organization key filter.
09	`search_users`	Searches for specific user accounts associated with the connected organization.

See It in Action

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

U Fetch the quality gate status of the main monolith backend project in SonarCloud.

Querying `get_quality_gate_status` for `monolito-backend-core`:
Status: ERROR 🔴

The Quality Gate failed because new code coverage dropped natively to 74% (against the 80% mandatory target) and 2 new code smells were introduced. Want me to dive deeper into `search_issues` to list the bad smells?

U List all registered organizations tied to my SonarCloud profile along with our connected users.

Running organizational discovery (`list_organizations`) and user lookup (`search_users`).

Organizations:
- 🏢 `vinkius-global-labs` (Name: VinkiusLabs LLC)
- 🏢 `personal-dev-sandbox` (Name: Renan Sandbox Hub)

Active Users under Global Scope:
- `renato_admin` (Active)
- `carlos_dev` (Active)
- `samuel_ci_bot` (Service Account)

Let me know if you need specific details for one logical group alone.

U Pull all the 'Major' and 'Critical' open issues for the API backend service codebase.

Accessing `search_issues` filtering dynamically for CRITICAL and MAJOR severities assigned to target `api-backend-srv`.

Detected Issues (Filtered View):

1. 🚨 **CRITICAL** (Bug): `NullPointerException` likely in `src/auth/tokenValidator.js` line 44 (Key: `AXb13k`)
2. 🚨 **CRITICAL** (Vulnerability): Hardcoded token visible in class `EmailMailer` at `src/services/mail.js` (Key: `AWz1L2`)
3. ⚠️ **MAJOR** (Code Smell): Cognitive complexity of `handleUserProcess()` exceeds tier 18 inside `controller.js` (Key: `AYx99m`)

Frequently Asked Questions

01 **How do I use SonarCloud MCP to find security vulnerabilities?**

Use `search_issues` and filter by 'CRITICAL' severity, then call `get_issue_details` on the found issue. This lets you pull exact details on hardcoded tokens or other exposed flaws.

02 **Can SonarCloud MCP tell me if my code coverage is enough?**

Yes. You use `get_project_measures`, specifying 'coverage' as the metric key, to pull the current percentage for any project component and compare it against your required minimum.

03 **What do I need to link SonarCloud MCP?**

You must provide a valid SonarCloud Security Token during setup. This is how your agent gains permission to read your organization's data.

04 **Does the SonarCloud MCP support multiple projects?**

Yes, you can use `search_projects` to find all available project keys and then run checks like `get_quality_gate_status` against each one individually.

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 `"sonarcloud": { "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

SonarCloud 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 SonarCloud. 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	SonarCloud MCP
Server ID	`019d760a-ff03-723e-b5f4-53f99a9b1dd3`
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/sonarcloud.