

M C P S E R V E R

N O C O D E

C L O U D H O S T E D

GitHub MCP for AI Agents

Manage your repositories and track pull requests through natural conversation.

GitHub MCP lets you manage your entire software development lifecycle through a chat interface. You can check the status of a pull request, list open issues, or search for specific code snippets without ever leaving your primary workspace. It gives your AI agent direct access to your repositories, making it easier to audit codebases or update project statuses on the fly.



Quality Score 100/100

version-control

repository-management

pull-requests

issue-tracking

code-search



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

GitHub MCP

18 tools available

Cloud-hosted on Vinkius

This MCP lets you manage your entire software development lifecycle through a chat interface. You can check the status of a pull request, list open issues, or search for specific code snippets without ever leaving your primary workspace.

Instead of jumping between your browser and your terminal, you can ask your agent to pull the contents of a configuration file or fetch repository metadata to see why a build is failing. It handles the tedious tasks like listing contributors or monitoring forks so you can stay focused on writing actual code. Because it's hosted on the Vinkius Marketplace, you get a reliable connection that keeps your source control data synced with your AI client. You can access your profile information, search through massive databases for specific code snippets, and manage your pull requests using plain English. It's about making your codebase searchable and actionable without the manual overhead of navigating the GitHub web UI. You can use it to audit a project's history, verify who is contributing to a specific repo, or even open new issues for bugs you find during a test run. It works by providing a bridge that translates your natural language requests into precise actions across your organization's projects. Whether you're looking for a specific line of code to understand a legacy system or you need to organize a messy backlog of bugs, this setup gives your agent the context it needs to be useful. It's about removing the friction of manual data entry and navigation, allowing you to move from an idea to a pull request faster than ever.

Core Capabilities

01 — Search for specific code snippets

Find specific logic blocks or code lines across your entire repository history.

02 — Open new issues or pull requests

Create new tasks or submit code changes directly from your chat window.

03 — Fetch file contents

Let your agent read the raw text of any file to help it understand your code.

05 — Track project progress

Monitor the status of pull requests and issues to keep your team on schedule.

04 — List user repositories

See all the repositories owned by a specific user or organization at a glance.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to the GitHub MCP on the Vinkius Marketplace.
- 02 Enter your GitHub Personal Access Token in the configuration settings.
- 03 Ask your AI client to list repositories or check issue statuses.

The bottom line is you get a direct line of communication between your AI agent and your GitHub data.

Built For

This is for the developer who spends too much time switching between browser tabs and the terminal. It's for anyone who needs to keep their project moving without the friction of manual navigation. It's built for people who want to stay in the flow state while managing the administrative side of code.

Software Engineer

Checking PR statuses or grabbing file contents during a debugging session on a Tuesday afternoon.

Tech Lead

Auditing repository metadata and contributor lists for new projects to ensure the right people are involved.

Project Manager

Getting a high-level overview of open issues and repository activity without logging into the dashboard.

What Changes When You Connect

- 01 Stop switching tabs by using `list_pull_requests` and `list_repo_issues` to see project status instantly.
- 02 Speed up debugging by using `get_file_contents` to let your agent read your code without manual copy-pasting.

-
- | | |
|-----------|--|
| 03 | Simplify project tracking with <code>create_new_issue</code> and <code>add_issue_comment</code> to manage tasks from your chat. |
| <hr/> | |
| 04 | Audit your codebase faster using <code>search_github_code</code> to find specific snippets across multiple repos. |
| <hr/> | |
| 05 | Get a bird's-eye view of your work with <code>list_user_repositories</code> and <code>list_org_repositories</code> to see what's active. |
-

Real-World Applications

Debugging a build failure

Ask your agent to `get_file_contents` for a config file to see why the environment isn't loading.

Sprint planning

Use `list_repo_issues` to summarize all high-priority bugs for the team during a standup.

Onboarding a new dev

Use `list_org_repositories` and `get_repository_details` to give a new hire a full overview of the project.

Code auditing

Use `search_github_code` to find every instance of a specific function across the entire organization.

Patterns to Avoid

Trying to host a website

X AVOID

Attempting to use this MCP to serve web content or host a live site.

✓ INSTEAD

Use a dedicated hosting provider for web deployment. This MCP is built for repository management and code interaction.

Manual copy-pasting

X AVOID

Copying and pasting large amounts of code into your chat to explain a bug.

✓ INSTEAD

Use `get_file_contents` so your agent can read the file directly from the repository.

Navigating to the dashboard

X AVOID

Clicking through multiple pages in the browser to find a specific pull request status.

✓ INSTEAD

Use `list_pull_requests` to see the status of all active changes in one command.

The Right Fit

Use this if you need your AI agent to interact with your source code, manage your issues, or search through your repositories. It's perfect for developers who want to automate the admin side of coding, like checking PR statuses or fetching file contents for debugging. It's a huge time saver for anyone who finds themselves constantly navigating the GitHub web interface to find basic information. Don't use this if you're looking to host a live website or manage a non-code project like a CRM; for those, you'd need a specific service connector. If you just want to read a few files once in a while, a simple manual download is faster. But if you want your agent to have a living, breathing connection to your codebase for ongoing tasks, this is the right choice. It bridges the gap between your high-level goals and the granular details of your repository data.

GitHub MCP for GitHub Repository Management

Most developers waste hours jumping between their terminal, the GitHub web UI, and various project management tools. You have to click through tabs to see if a PR is merged, copy-paste code into a chat to explain a bug, and manually update issue statuses. It's a constant context switch that breaks your flow.

This MCP puts your entire repository lifecycle into your chat interface. You can ask your agent to summarize all open issues, check the contents of a specific file, or create a new pull request with a single command. You get a unified workspace where your code and your tasks live together.

GitHub MCP for GitHub Code Search

Finding a specific logic block or a hidden configuration across dozens of repositories usually involves endless scrolling and complex search queries. You have to remember which repo contains what and hope the search results actually show you the right line of code.

The GitHub MCP lets you perform deep searches across the entire database instantly. You can isolate specific snippets and have your agent analyze them immediately. It turns a 20-minute search into a 5-second conversation.

GitHub MCP with 18 Tools for Repository Management

Use these tools to let your AI agent manage issues, search code, and track pull requests across your GitHub account.

#	TOOL	DESCRIPTION
01	verify_api_connection	Check if your connection to GitHub is active and working. This ensures your agent can access your data without errors.
02	list_user_repositories	See all the repositories owned by a specific user account. It helps you get a quick overview of your personal projects.
03	create_pull_request	Create a new pull request by specifying the head and base branches. This lets you submit changes without leaving your chat.
04	add_issue_comment	Post a new comment to an existing issue or pull request with markdown support. You can provide feedback or updates directly to your team.
05	create_new_issue	Open a new issue in a repository to track a bug or feature request. This helps you organize your work and stay on top of tasks.
06	get_file_contents	Pull the raw text content of a specific file from a repository. Your agent can use this to understand your code structure.
07	get_issue_details	Get full information about a specific issue using its unique number. This gives you all the context you need for a specific bug.
08	get_my_github_profile	Retrieve the identity and profile details of the authenticated user. Use this to verify permissions and account context.
09	get_repository_details	Fetch metadata and configuration details for a specific repository. This provides the high-level info needed for project auditing.
10	list_branches	See all the different branches available within a repository. Use this to navigate different versions of your code.
11	list_commits	View the most recent commits, with options to filter by branch or SHA. This helps you track the latest changes made to the code.
12	list_repo_issues	Get a list of all issues currently open or closed in a repository. Use this to see what tasks are currently in progress.
13	list_org_repositories	See every repository owned by a specific organization. This is great for getting a bird's-eye view of a company's work.

#	TOOL	DESCRIPTION
14	list_pull_requests	View all current pull requests for a repository. This lets you see what's waiting for review at a glance.
15	list_releases	See the release history and versions for a project. Use this to track the deployment timeline of your software.
16	list_stargazers	See who has starred a repository to track its popularity. This helps you monitor community engagement and project reach.
17	search_github_code	Search for specific code snippets across the entire GitHub database. Use this to find logic blocks buried deep in your codebase.
18	search_github_repositories	Find specific repositories based on your search terms. This helps you locate projects across a large organization.

See It in Action

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

U What's the status of the 'login-fix' pull request?



Pull Request Status

PR: login-fix

- **Status:** ● Open
- **Author:** @dev_jane
- **Files Changed:** 4
- **Reviewers:** 2

Would you like to see the specific comments on this PR?

U Find all the open issues in the 'frontend-app' repo.



ISSUE #	TITLE	LABEL
102	Fix header overlap	Bug
105	Update logo assets	Task
110	Add dark mode toggle	Feature

There are 3 open issues. Should I fetch the details for any of these?

U Show me the contents of the .env.example file in the 'backend' repo.



File Content: .env.example

```
`` bash
DB_HOST=localhost
DB_PORT=5432
API_KEY=your_key_here
DEBUG=True
``
```

Let me know if you need me to check any other files in the backend folder.

Frequently Asked Questions

01 Can the GitHub MCP manage my pull requests?

Yes, it lets your agent list, track, and create pull requests so you don't have to leave your chat.

02 How does the GitHub MCP help with debugging?

You can use it to fetch file contents directly, allowing your agent to see your code and suggest fixes.

03 Can I use the GitHub MCP to search for code?

Yes, it can search for specific code snippets across your repositories to help you find what you need faster.

04 Does the GitHub MCP work for private repositories?

It works with any repository you have access to, provided you use a valid personal access token.

05 Can I use the GitHub MCP to create issues?

You can open new issues and add comments to existing ones directly through your AI client.

06 How do I connect my GitHub account?

You just need to provide a Personal Access Token in your configuration after subscribing on Vinkius.

07 How do I get a Personal Access Token (PAT) for GitHub?

Log in to GitHub, go to Settings > Developer settings > Personal access tokens, and you can generate a new token (classic or fine-grained) there.

08 Can I read private repositories?

Yes, provided that your Personal Access Token has the necessary scopes (e.g., 'repo') to access your private content.

09 How do I search for code within a specific organization?

Use the 'search_github_code' tool and include the organization qualifier in your query, for example: 'org:vinkius my-term'.

10 Can I comment on issues via the agent?

Yes! Use the 'add_issue_comment' tool — provide the repository owner, name, and the issue or PR number along with your comment text (markdown supported). Your agent will post the comment instantly.

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 <code>"github": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

GitHub 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 GitHub. 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	July 2026
MCP Server	GitHub MCP
Server ID	019d75a5-5797-71cd-bcb0-5c00a4eb4d74
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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/github.