MCP SERVER | NO CODE | CLOUD HOSTED

GitHub MCP

Audit commits, track PRs, and manage issues from chat.

Manage your entire development lifecycle with this MCP. Your AI agent connects directly to your GitHub account, letting you review pull requests, audit commit history, list open issues, and monitor CI/CD workflows—all through conversation. Stop context switching; start coding.

A+ Quality Score 100/100

version-control | code-review | ci-cd | repository-management | issue-tracking | workflow-automation

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

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

GitHub MCP

14 tools available
Cloud-hosted on Vinkius

Need to know what's happening in a repo without opening the browser? This MCP gives your AI agent direct access to your entire GitHub account. You can talk to it like an engineering lead, getting status updates on branches, tracking pull requests, or auditing commits across different teams. It lets you list all repositories for the user and even search them using advanced qualifiers like language type or star count. Want to start a new task? You can create issues with specific labels and markdown descriptions, or check recent GitHub Actions runs to see if the latest build failed. By connecting through Vinkius's catalog, your AI client becomes a single point of truth for every dev workflow, letting you focus on code, not tabs.

Core Capabilities

01 — Audit repository status

Get details about specific repositories, list all available branches in a project, and search across your entire portfolio using advanced query syntax.

02 — Manage development tasks

Create new issues or inspect existing ones by retrieving their number, title, labels, and assignment status.

03 — Track code changes and reviews

Review pull requests for merge status, view commit logs across branches, and list all releases attached to a specific tag.

04 — Monitor automated workflows

Check the recent runs of GitHub Actions pipelines to determine if deployments succeeded or failed.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your GitHub Personal Access Token.

02 Connect it to any AI-compatible client, like Cursor or Claude.

03 Ask your agent a question—like 'Show me all open issues in the backend repo'—and get structured data back instantly.

The bottom line is that you never have to leave your chat interface or IDE to manage development tasks again.

Built For

This MCP is for the software developer, the tech lead, and the operations engineer who spends more time navigating dashboards than writing code. If checking repo status or tracking PRs feels like a manual chore, this is built for you.

Software Developer

Uses the MCP to quickly check issue status, review pull requests, and inspect commit logs without switching away from their IDE.

Engineering Manager

Audits open issues across multiple teams, tracks release progress by fetching specific releases, and monitors CI/CD health.

DevOps Engineer

Checks recent GitHub Actions workflow runs to verify deployment statuses and uses the MCP to manage repositories programmatically.

What Changes When You Connect

01 Stop context switching: You don't need to open GitHub.com or switch tabs; your agent handles the data retrieval directly in the chat window.

02 Full lifecycle visibility: Quickly check commit logs using list_commits, review PRs with get_pull_request, and monitor CI/CD health via list_workflow_runs all from one prompt.

03 Advanced repository discovery: Use search_repos to find highly-starred or language-specific repos across your organization's entire catalog instantly.

04 Issue management at scale: List issues and create new ones using create_issue, ensuring every bug report has the right labels and owner assigned.

05 Auditability for managers: Get a comprehensive view of all releases with list_releases and check repository details with get_repo, making audits simple.

Real-World Applications

Triage a major bug report

A maintainer gets a critical issue ticket. They ask their agent to use list_issues filtered by 'bug' and then check the last commits using list_commits on that repo. The agent identifies the relevant code segment, allowing the maintainer to assign an owner immediately.

Review cross-team feature integration

The tech lead needs to know if two different teams' features can coexist before merging. They ask their agent to get_pull_request for both branches and then check the list_workflow_runs status, ensuring all necessary CI/CD tests passed.

Find a suitable project boilerplate

A new developer needs a starting point. Instead of browsing manually, they ask their agent to search_repos for 'Python' with stars greater than 10k and language:python. The results provide instant, actionable suggestions.

Track compliance or version changes

The ops engineer needs to confirm which version of the service was deployed last month. They ask their agent to list_releases using a date qualifier, instantly verifying the exact git tag and associated assets.

Patterns to Avoid

Using manual API calls

✗ AVOID

Writing complex curl commands or building out webhooks just to check if a PR was merged or what commits happened last week.

✓ INSTEAD

Just ask your agent. Use get_pull_request for status checks, or list_commits on the branch name you care about.

Relying only on web search

✗ AVOID

Searching Google for 'GitHub repo management' and finding generic articles instead of real-time data like open issues or failing builds.

✓ INSTEAD

Connect this MCP. Your agent accesses live, structured data from the GitHub API directly in your chat.

Ignoring repository scope

✗ AVOID

Trying to check a repo that's private but forgetting to provide the correct owner or using outdated tokens.

✓ INSTEAD

Use get_user first to verify your token works, then specify the full owner and repo name in any tool call.

The Right Fit

Use this MCP if your job involves managing the development lifecycle: reviewing PRs, tracking bugs, monitoring deployments, or auditing code history. You need an AI agent to act as a proxy for navigating GitHub's complex UI and API endpoints.

Don't use it if you just need simple document storage or basic CRUD operations outside of Git (e.g., managing tickets in Jira). If your task is purely about writing documentation, stick with dedicated text editors. This MCP is strictly for version control and code review data. Always remember that while tools like list_issues give you the raw data, you still need to interpret it based on project context.

The Problem of Context Switching

Today, checking a single feature's status means jumping through hoops. You open GitHub.com for the PR status; then you navigate to the Issues tab to check bug reports; after that, you switch to the Actions tab just to see if the latest build failed. It’s three different tabs, five clicks deep, and half your time is spent clicking instead of thinking.

With this MCP, it's a single conversation. You talk to your agent—saying 'What's the status of PR #142 in the billing repo?'—and you get all that information back immediately. Your AI acts like an experienced dev lead who already knows where everything is.

Get Status Updates with list_workflow_runs

Manually checking the status of a deployment requires navigating to the Actions tab and sifting through logs. You have to scroll back, figure out which commit triggered it, and determine if the 'build Docker image' step failed or just timed out.

Now, you simply ask your agent to list_workflow_runs for that repo. It gives you a clean summary of the last five runs, telling you exactly what succeeded and where the failure happened—no scrolling necessary.

GitHub Alternative: 14 Tools for Dev Ops

Use these tools to programmatically list repositories, create new issues, check commit history, review pull requests, and manage workflow runs through natural conversation.

#	TOOL	DESCRIPTION
01	`create_issue`	Generates a brand new GitHub issue with specified title, body, and labels.
02	`get_issue`	Retrieves all details for a single GitHub issue using its number.
03	`get_user`	Verifies your connection by retrieving basic information about the authenticated GitHub user.
04	`get_pull_request`	Fetches detailed status and review history for a specific pull request number.
05	`get_release_by_tag`	Retrieves full release details, including assets and publish dates, based on a git tag name.
06	`get_repo`	Gets detailed information about any specific GitHub repository by owner and name.
07	`list_branches`	Lists all branches within a repository, noting which one is the default or protected branch.
08	`list_commits`	Retrieves a list of commits on a specified branch, showing the author, message, and date for each.
09	`list_issues`	Lists all issues in a repo, allowing filtering by state like open or closed.
10	`list_pull_requests`	Provides an overview of pull requests, including their merge status and draft status.
11	`list_releases`	Lists all releases associated with a repository in reverse chronological order.
12	`list_repos`	Displays a list of repositories owned by the authenticated user, showing details like language and star counts.
13	`list_workflow_runs`	Lists recent GitHub Actions workflow runs for a repository, including their final status (success or failure).
14	`search_repos`	Searches across all repositories using powerful qualifiers like language type or star counts.

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 open issues in my backend-api repo.

I found 7 open issues in backend-api. The most critical is #142 'Database connection timeout on high load' (bug, high-priority, assigned to @alice). Other notable issues include #138 'Add rate limiting middleware' and #135 'Refactor auth middleware to support OAuth2'.

U What's the status of recent workflow runs on main branch?

The last 5 workflow runs on main: CI Pipeline (success, 12m ago), Deploy to Staging (success, 10m ago), Security Scan (success, 8m ago), Release Build (failure, 2h ago — failed at 'build Docker image' step) and Integration Tests (success, 3h ago).

U Search for popular Python machine learning repos with over 1000 stars.

I found several highly-starred Python ML repos: huggingface/transformers (119k stars), scikit-learn/scikit-learn (58k stars), tensorflow/tensorflow (185k stars), pytorch/pytorch (79k stars) and microsoft/ML-For-Beginners (27k stars). Would you like more details on any of these?

Frequently Asked Questions

01 **How do I check if my account token is working with get_user?**

Run the get_user tool first. It retrieves your login name, company, and follower count, confirming that your MCP connection and Personal Access Token are valid.

02 **Can list_issues filter by assignee?**

Yes, the list_issues tool returns details about who is assigned to the issue. You can prompt your agent to specifically find issues assigned to a certain team member.

03 **Is there a way to search for repos across all my accounts? Use search_repos.**

The search_repos tool supports powerful qualifiers, allowing you to filter by language, star count (e.g., 'stars:>1000'), or organization name.

04 **What is the difference between list_issues and list_pull_requests?**

list_issues shows general bugs and feature requests. list_pull_requests focuses specifically on code review items, providing details about draft status and merge readiness.

05 **How can I check the history of a specific branch? Use list_commits.**

You use list_commits by specifying the owner, repo, and the target branch name. The results provide the SHA, author, message, and date for every commit.

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 `"github-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

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	June 2026
MCP Server	GitHub MCP
Server ID	`019d8440-921f-7143-9eae-3f36b3c6b0f6`
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/github-alternative.