

MCP SERVER

NO CODE

CLOUD HOSTED

Gitee MCP

Manage Code, Issues, and PRs via Conversation

Gitee MCP gives your AI client full command over your collaborative code base. Stop clicking through web interfaces to manage repos, track bugs, or review pull requests. This connector lets you ask natural language questions—like 'What commits were made on the staging branch last week?' or 'Create a new issue for this broken login flow'—and get instant, actionable data back from Gitee.

A+ Quality Score 100/100

git-repository

code-hosting

issue-tracking

pull-requests

version-control

software-development



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.

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.

Gitee MCP

10 tools available
Cloud-hosted on Vinkius

Dealing with large codebases means jumping between tabs: checking repo status here, opening an issue tracker there, and finally reviewing pull requests somewhere else. That friction slows you down.

This MCP connects your AI client directly to Gitee's core functions. Instead of navigating the web interface, you talk to your agent. Your agent reads repository data, lists all projects belonging to a user, and even generates new repositories on demand. Need to track something? You can list existing issues or create brand-new ones with just a prompt. When code changes, you don't have to manually audit the history; your agent pulls commit logs and details for specific files right into your conversation. This ability to treat complex development processes like a natural dialogue makes managing dev workflows simple. It's exactly what Vinkius built this catalog for: giving power users direct access to deep service functionality through conversational AI.

Core Capabilities

01 — View and Manage Code Repositories

The agent can list all your user repositories, get detailed project information, and create entirely new ones.

03 — Manage Code Changes

The system monitors pull requests for review and allows you to initiate the creation of a new pull request directly from your prompt.

02 — Track Bugs and Features

You can view lists of repository issues or create a new issue to track bugs without leaving your workspace.

04 — Audit Code History

You can retrieve comprehensive lists of commit logs or read specific file contents using just descriptive language.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP and provide your Gitee Personal Access Token (PAT) within Vinkius.
- 02 Connect your preferred agent, like Cursor or Claude, to the Vinkius platform.
- 03 Ask your AI client to perform a development task—for example, 'List all repos for me' or 'Create an issue detailing X bug.' — and get the result instantly.

The bottom line is you use natural language commands through your agent rather than clicking buttons in a web dashboard.

Built For

Anyone who spends more time managing development processes than actually writing code. This MCP solves the pain of context switching and navigating dozens of tabs to complete simple tasks.

Software Developer

Uses this to list user repositories, retrieve file contents for debugging a specific module, or quickly create an issue when they hit a roadblock.

Project Manager

Leverages it to monitor pull requests and audit issues across multiple projects without needing permission to log into every single repository dashboard.

DevOps Engineer

Uses it to integrate Gitee data into automated workflows, checking commit history or creating new repositories as part of deployment routines.

What Changes When You Connect

- 01 Instead of jumping to the issue tracker page, simply ask your agent to list repository issues or create a new one. This makes bug reporting immediate and contextual.

-
- 02 You can perform advanced code audits without opening multiple tabs. Just prompt for commit logs or file contents; the data appears instantly in your chat window.

 - 03 Need a new project? Use the tool that creates repositories so you don't have to manually navigate the creation forms on the web platform.

 - 04 Reviewing code changes is faster than ever. The ability to list pull requests means you can check status and initiate reviews without leaving your development environment.

 - 05 This MCP lets you manage everything from listing user repositories to getting detailed repo information, giving you a single pane of glass for all your projects.
-

Real-World Applications

The onboarding developer needs project context

A new dev asks their agent: 'What's the current state of the payment module?' The agent calls `get_repo` and then `gets_file_content` for key files, summarizing which parts need attention. They didn't have to read a massive README file.

The team needs to propose a new feature

A developer prompts: 'I need a place for the new reporting tool code.' The agent uses `create_repo`, generating the necessary project structure instantly and providing the link.

The PM needs to track down legacy bugs

A manager asks the agent: 'Show me all issues related to user authentication created last quarter.' The agent uses `list_repo_issues`, filters by date/label, and presents a clear report without needing manual filtering.

The core dev is merging changes and needs to review them

Instead of manually opening the PR list, they ask: 'List all open pull requests for the main branch.' The agent calls `list_pull_requests`, giving an immediate status overview.

Patterns to Avoid

Trying to view a file's content

✗ AVOID

Manually navigating to the 'src/utis' folder in the web UI and then clicking into a specific README file just to copy text.

✓ INSTEAD

Just tell your agent: 'Get the file content for src/utis/README.md.' The tool handles the pathing, saving you multiple clicks.

Creating a new project

✗ AVOID

Logging into Gitee's main page and clicking through three different menus to find the 'New Repository' button.

✓ INSTEAD

Tell your agent: 'Create a repo called inventory-service.' It uses create_repo immediately.

Tracking down bug details

✗ AVOID

Searching for an issue using the keyword in the web search bar, only to find 50 irrelevant results that require manual sifting.

✓ INSTEAD

Tell your agent: 'List all open issues related to checkout payments.' The tool uses list_repo_issues and provides a focused, actionable list.

The Right Fit

Use this MCP if your workflow relies on synthesizing information across multiple development areas—like cross-referencing commit logs with current repository issues. It's best for developers who operate conversationally and need deep access to code history or project structure details.

Don't use it if you just need a simple webhook trigger (e.g., 'When X happens, send an email'). For those limited triggers, a direct API integration might be cleaner. However, if your goal is *information retrieval* and *action* based on that retrieved data, this MCP is perfect because it turns the entire platform into a conversational source of truth. If you're only listing repositories without needing details or actions (like creating issues), you can use simpler category tools, but this provides the full lifecycle management.

The friction comes from context switching.

Right now, updating your project status is a multi-step chore. You have to log into Gitee, check the 'Issues' tab for bugs, then switch over to the 'Commits' tab to see when someone fixed it, and finally jump to 'Pull Requests' to make sure the fix didn't break anything else. It's clicking, copying, pasting, and mentally tracking where you left off.

With this MCP, that entire process collapses into a single dialogue. You ask your agent about the bug, and in one response, it synthesizes data from issues, commit logs, and pull requests. You get an immediate, coherent answer without ever leaving your chat window.

Gitee MCP gives you conversational repository control.

The manual steps that vanish are the repeated logins, the navigating between tabs (Issues -> Commits -> PRs), and the need to copy a repo name just for an API call. Your agent handles all of that under the hood.

It's not just about listing things; it's about receiving synthesized knowledge immediately. You finally get to treat your entire development stack like one conversation.

Gitee: 10 Tools for Code Development

Use these tools to interact with every core function of Gitee, allowing your AI client to perform actions like creating issues or listing commits directly.

#	TOOL	DESCRIPTION
01	<code>get_user</code>	Retrieves the profile information for the authenticated user on Gitee.
02	<code>list_pull_requests</code>	Shows a list of existing pull requests associated with repositories.
03	<code>list_repo_commits</code>	Provides a log detailing commits made within a specific repository.
04	<code>list_repo_issues</code>	Generates a list of bugs or feature requests logged for a repository.
05	<code>list_user_repos</code>	Lists all repositories owned by the currently authenticated user account.
06	<code>create_issue</code>	Creates and logs a brand new issue detailing a problem or suggestion in a repository.
07	<code>create_pull_request</code>	Initiates a pull request to merge changes between branches.
08	<code>create_repo</code>	Creates an entirely new, blank repository account on Gitee.
09	<code>get_file_content</code>	Fetches and reads the actual text content of a specified file path in a repository.
10	<code>get_repo</code>	Retrieves general structural details about a specific repository.

See It in Action

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

U List all my repositories on Gitee.



I've retrieved your Gitee repositories. You have 5 active projects, including 'awesome-project' and 'demo-app'. Would you like more details on any of them?

U Create a new issue in 'my-org/my-repo' titled 'Bug: login failure'.



Done! I've created issue #42 in the 'my-org/my-repo' repository with the title 'Bug: login failure'. You can track it here: [link].

U Read the contents of README.md from the main branch of 'user/project'.



I've retrieved the README.md content for you. It contains project documentation, installation steps, and usage examples. Would you like me to summarize it?

Frequently Asked Questions

01 How do I list my user repositories using Gitee MCP?

You simply ask your agent to list all user repositories. The system runs the `list_user_repos` tool and returns a comprehensive list of every project you own.

02 Can I create an issue in Gitee via the Gitee MCP?

Yes, you can. Just ask your agent to 'Create a new issue' in a specific repository, detailing the title and body content for bug tracking.

03 What if I need to see old code history with Gitee MCP?

You prompt the agent to list repository commits. It executes `list_repo_commits`, showing you a detailed log of changes across your specified branches and projects.

04 Does Gitee MCP help me review pull requests?

Yes, you can use `list_pull_requests` to see the status of open or closed PRs. You can even ask the agent to create a new pull request for merging code.

05 Is Gitee MCP only for large teams?







No. While it's powerful for teams, solo developers use it constantly too. It lets you list your own repositories and manage personal projects just as easily.

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>"gitee": { "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

Gitee 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](https://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 Gitee. 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	Gitee MCP
Server ID	019d8440-64ce-713c-af21-9d08b43dc3af
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/gitee.