

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 Gists MCP

Manage your code snippets and configs directly through your AI agent.

GitHub Gists MCP lets your AI client interact with your GitHub Gists. You can browse public snippets without any setup, or provide a GitHub token to let your agent create, edit, fork, and delete Gists on your own account. It turns your agent into a specialized tool for managing code fragments, configuration files, and scripts.

A+ Quality Score 96.67/100

github

gists

code-snippets

code-sharing

version-control

scripts



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 Gists MCP

8 tools available

Cloud-hosted on Vinkius

You can now treat GitHub Gists as a native part of your AI agent's workspace. Instead of manually copying and pasting code, you can tell your agent to find a specific snippet, read a configuration file, or publish a new script directly to your GitHub account. If you just need to grab something public, you don't even need to provide credentials. If you want to manage your own private snippets, just add a GitHub token. This MCP handles the heavy lifting, including streaming large files over 1 MB and providing clear retry hints if you hit GitHub's rate limits. Whether you are pulling a Docker compose file to inspect it or forking a teammate's snippet to make your own version, your agent handles the API calls while you focus on the code.

Core Capabilities

01 — Public Discovery

Your agent can browse and read public Gists without any GitHub credentials.

02 — Private Management

With a GitHub token, your agent can access and edit your private Gists.

03 — File Streaming

The MCP streams files larger than 1 MB directly from their raw URLs to your agent.

04 — Content Modification

Your agent can update descriptions, change visibility, or wipe file contents in a single call.

05 — Version Control Lite

You can fork existing Gists to create independent copies for your own use.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/en/ai-agent-connect/github-gists — connect your AI agent in three steps.

- 01

Subscribe to the GitHub Gists MCP on Vinkius.
- 02

Provide a GitHub token if you need to manage private Gists or write data.
- 03

Connect your preferred client like Claude, Cursor, or Windsurf.
- 04

Ask your agent to find, read, or publish Gists using natural language.

Setting up this MCP takes seconds and works with any compatible client.

Built For

This MCP is built for developers and engineers who use Gists to store small, reusable pieces of code and configuration.

Developers

They hand off the task of organizing and updating code snippets and notes to their agent.

Platform Engineers

They use the agent to programmatically surface shared configuration snippets or documentation.

Automation Builders

They let their agent publish and distribute versioned scripts via Gists.

What Changes When You Connect

- 01

Read public code snippets without setting up GitHub credentials.
- 02

Manage private Gists using a single GitHub token.
- 03

Handle large files through automated raw URL streaming.

04 Get clear retry hints when hitting GitHub rate limits.

Real-World Applications

Snippet Management

Ask your agent to save a new Python function as a secret Gist for later use.

Config Inspection

Have your agent read a specific YAML config file from a public Gist to verify settings.

Code Iteration

Fork a public bash script to your own account so you can modify it safely.

Rapid Deployment

Use your agent to publish a new deployment script to GitHub in seconds.

Patterns to Avoid

Manual copy-paste loops

❌ AVOID

You find a useful script on a public Gist and manually copy the text into your editor to make changes. This wastes time and breaks your flow.

✓ INSTEAD

Let your agent handle the heavy lifting. Use ``fork_gist`` to create a copy in your own account so you can edit it immediately.

Blindly deleting snippets

❌ AVOID

You tell your agent to clean up old files without checking what they are first, accidentally wiping out a critical config.

✓ INSTEAD

Always check the contents first. Use ``get_gist`` to inspect the files before you decide to run ``delete_gist``.

Ignoring rate limits

❌ AVOID

Your agent tries to pull hundreds of public Gists in a single loop, hitting GitHub's limits and failing.

✓ INSTEAD

The MCP handles this by providing retry hints. If you hit a limit, wait for the hint and try again.

The Right Fit

Connect this MCP if you frequently use Gists to store reusable code, config files, or deployment scripts. Don't connect it if you only use GitHub for full repositories. The decision comes down to whether you want your agent to treat small snippets as a live, editable workspace.

GitHub Gists Management Tools

A suite of tools for reading, creating, and managing code snippets and configuration files directly through your AI client.

#	TOOL	DESCRIPTION
01	fork_gist	This tool creates a copy of a Gist in your own account. Use it to modify someone else's code without changing the original.
02	create_gist	This tool publishes new Gists to GitHub. You can send one or multiple files and choose to keep them secret or make them public.
03	delete_gist	This tool permanently removes a Gist and all its files. It requires a token and you should confirm before your agent runs it.
04	get_gist	This tool fetches a specific Gist and its contents. Public Gists are accessible without a token, but private ones require authentication.
05	list_user_gists	This tool retrieves a list of Gists belonging to a specific GitHub user. Providing a token allows your agent to see private Gists as well.
06	get_gist_file	This tool reads a single file from a Gist as plain text. It handles large files by streaming them from their raw URL.
07	list_public_gists	This tool lets your agent browse recently updated public Gists across all of GitHub without needing any credentials.
08	update_gist	This tool modifies an existing Gist. You can use it to change the description, toggle visibility, or update file contents.

See It in Action

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

U List the most recently updated public Gists and tell me what each one is about.



I listed the 10 most recently updated public Gists. They include a CI pipeline tweak, a Docker compose snippet, a SQL migration and a few Python utilities. The top ones: owner-a/abc (Kubernetes config, updated today) and owner-b/def (bash deploy script). Want me to open any of them in full?

U Create a private Gist from this bash script with the description "deploy helper".



I created a secret Gist "deploy helper" with your script (2.1 KB, deploy.sh). It is not public. You can make it public or add more files with update_gist whenever you like.

U Fork this Gist so I can edit my own copy without touching the original.



I forked the Gist to your account (new id ...). It has 4 files: main.py, config.json, README.md and test_main.py. You can now edit your copy, the original stays unchanged.

Frequently Asked Questions

01 Do I need a GitHub token to use this MCP?

No, you can use it keyless to read public Gists. You only need a token to access private Gists or to create, edit, and delete Gists.

02 What is the rate limit for keyless requests?

Without a token, you are limited to about 60 requests per hour per IP. With a token, you get an authenticated budget of about 5000 requests per hour.

03 Can my agent create public Gists?

Yes, when using the create tool, you can set the visibility to public.

04 How does the MCP handle large files?

For files larger than 1 MB, the MCP streams the content directly from the raw GitHub URL to ensure your agent receives the full file.

05 Is deleting a Gist permanent?

Yes, deleting a Gist is irreversible. The agent should confirm with you before executing a delete command.

06 Do I need a GitHub token to use this?

No. Public Gists can be listed, read and opened as files without any credential. A token is only required for your private Gists and for write operations (create, update, fork, delete).

07 What permissions does the token need?

Only the "gist" permission — a classic personal access token with gist scope, or a fine-grained token with access to Gists. No repo access is required. Create the token under Settings > Developer settings > Personal access tokens.

08 Can I read large Gist files in full?

Yes. The Gist payload includes up to 1 MB of content per file; `get_gist_file` transparently downloads larger files in full from their raw URL. For Gists whose files exceed 10 MB, clone them with the git URL (`git_pull_url`) returned by `get_gist`.

09 How does rate limiting work?

Keyless (public reads) is limited to about 60 requests/hour per IP; with a token the budget is about 5000 requests/hour. When a limit is hit the tool returns a clear error with the time to wait and when the budget resets, instead of failing silently.

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-gists": { "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 Gists 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 Gists. 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	September 2026
MCP Server	GitHub Gists MCP
Server ID	01a0d9c9-cc81-7294-87da-0b9a92c34ccb
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/en/ai-agent-connect/github-gists.