MCP SERVER | NO CODE | CLOUD HOSTED

ZenHub MCP

Track Epics, Boards & Estimates with Natural Language

ZenHub MCP connects your AI agent directly to ZenHub's project data. You can use natural language commands to view entire development boards, track epic progress, update issue statuses, and set story point estimates without leaving your IDE or chat client.

A+ Quality Score 100/100

agile-management | issue-tracking | sprint-planning | github-integration | kanban-boards | epics

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/zenhub

SCAN TO VIEW

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.

ZenHub MCP

8 tools available
Cloud-hosted on Vinkius

Your agent talks directly to ZenHub, giving it visibility into everything from the project board status to how many story points an epic needs. Instead of opening 15 tabs just to check on a sprint's health, you simply ask your AI client for a summary. It pulls in all relevant data—like which issues are stuck between 'In Progress' and 'Review/QA' or what the total estimate is for a large feature set. This MCP lets you manage complex agile boards and track release progress using plain conversation. Once connected through Vinkius, your agent becomes an embedded project assistant, making board visibility and status changes feel like talking to a coworker instead of logging into a dashboard.

Core Capabilities

01 — View Board Status

The MCP retrieves the entire ZenHub board view for any specified repository or workspace.

02 — Adjust Issue Workflow

You can instantly move issues between different pipelines to reflect their current development status.

03 — Set Task Estimates

The system allows you to assign or retrieve story point estimates for specific project issues.

04 — Analyze Epic Details

You get a full rundown of any ZenHub epic, including all the smaller constituent issues it contains.

05 — Monitor Project Releases

Retrieve comprehensive release reports and progress metadata for the entire project cycle.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and enter your ZenHub API Token.

02 Connect your preferred AI client (like Claude or Cursor) to the service.

03 Ask your agent a question, like 'Show me all issues in the Backlog pipeline,' and get an instant, structured answer.

The bottom line is you manage complex project data through conversational commands, not clicks.

Built For

This MCP is for anyone whose day involves tracking progress across multiple interconnected projects. If your job requires correlating status updates, estimates, and board movements across different teams, this tool saves hours of manual dashboard refreshing.

Project Manager

You monitor overall board health and epic progress by asking for summaries rather than digging through multiple reports.

Scrum Master

You facilitate planning sessions by querying release reports and checking team velocity metrics on demand.

Software Engineer

You update issue statuses and set estimates directly from your IDE without switching context or leaving your coding flow.

What Changes When You Connect

01 Stop manually clicking through boards. You can use natural language to view the entire ZenHub board status for a repository or workspace instantly.

02 Update issue progress without leaving your IDE. Commands allow you to move issues between pipelines immediately after coding them, keeping the workflow accurate.

03 Keep project estimates current. Use the tool to set or retrieve story point estimates for any issue, eliminating manual spreadsheet updates.

04 Gain full visibility into large features. List and inspect ZenHub epics to understand the scope of work before committing resources.

05 Get a clear picture of project history. Access release reports and progress metadata so you can track what was delivered and when.

Real-World Applications

A PM needs an immediate status report on a major feature.

The Project Manager asks their agent: 'What's the current state of the Q3 onboarding epic?' The agent uses `list_repo_epics` and `get_epic_data`, compiling a single answer detailing all constituent issues, which are currently stuck in the 'Review/QA' pipeline.

A developer finishes code and needs to move the ticket.

The Software Engineer runs: 'Move issue #45 from In Progress to Review.' The agent uses `move_issue_between_pipelines`, instantly updating the board status without requiring them to open the ZenHub UI.

Scrum Master needs to estimate a new set of tasks.

Instead of gathering team members, the Scrum Master asks: 'What are the estimated points for these three related issues?' The agent uses `get_zenhub_issue_data` and calculates the total needed capacity.

Tracking a multi-stage release cycle.

The PM queries: 'Show me the completion status of the last two releases.' The agent calls `list_release_reports`, providing key dates, progress metrics, and overall completion percentages immediately.

Patterns to Avoid

Checking board status via GitHub UI

✗ AVOID

Spending 10 minutes navigating the repository's Issues tab, filtering by pipeline (In Progress) and then manually checking each issue's status.

✓ INSTEAD

Ask your agent to use `get_repo_board` or `get_workspace_board`. This gives you a summarized board view in seconds, telling you exactly which pipelines are backed up.

Updating estimates in a spreadsheet

✗ AVOID

A developer finishes work and has to stop coding, open the ZenHub UI, find the issue, then manually change the estimate field.

✓ INSTEAD

Your agent handles this. Just tell it: 'Set the story point estimate for Issue #123 to 5.' The tool `set_issue_estimate` executes the update directly.

Finding out what an epic contains

✗ AVOID

The PM opens a high-level epic card, but then has to click into it multiple times and manually count all related tasks to understand the scope.

✓ INSTEAD

Ask your agent to use `list_repo_epics` followed by `get_epic_data`. It gives you a complete breakdown of every issue under that epic in one go.

The Right Fit

Use this MCP if your project planning relies heavily on real-time, interconnected data across boards, epics, and issues. If you need to know 'where' an issue is or 'how much' it costs (in points), this is the right tool. Don't use this if you just need a simple message notification; those are messaging tools. Also, don't use it if your workflow doesn't involve established pipelines and epics—it's highly specialized for agile board management. If you only need to pull raw issue descriptions from GitHub without status context, other generic data retrieval tools will work better.

The Manual Chore of Tracking Project Progress

Right now, keeping track of a project's health means jumping between multiple tabs and services. You open the main GitHub board, then switch to the issue tracker to check estimates, and finally toggle over to the epic view just to see if all related tasks are finished. It’s constant context switching and copy-pasting status updates into chat.

With this MCP, you keep everything in one place. You simply ask your agent, 'What's the status of the payment processing feature?' The answer isn't a link; it's a synthesized report that shows the epic name, how many issues are complete, and which ones are stalled in QA—all delivered conversationally.

Getting ZenHub Data with the ZenHub MCP

The manual steps that disappear include navigating to `get_workspace_board` just to see a high-level overview, or having to call the API manually to check if an issue was successfully moved from 'Backlog' to 'In Progress'. You don't write the query; you just ask.

You get instant, actionable intelligence. It’s not just data retrieval; it's context delivered directly into your workflow, letting you make decisions without ever opening a dashboard.

ZenHub MCP with 8 Tools

These eight tools allow you to programmatically interact with every aspect of your agile project management setup, from board views to individual issue details.

#	TOOL	DESCRIPTION
01	`get_repo_board`	Retrieves the complete ZenHub board view for a specific repository.
02	`get_epic_data`	Fetches detailed information about a specified ZenHub epic.
03	`list_repo_epics`	Generates a list of all available ZenHub epics within a repository.
04	`set_issue_estimate`	Updates the story point estimate for any given issue in ZenHub.
05	`get_zenhub_issue_data`	Pulls ZenHub-specific metadata about a particular GitHub issue.
06	`move_issue_between_pipelines`	Changes an issue's pipeline status, moving it between different stages of the workflow.
07	`list_release_reports`	Generates a list of comprehensive release reports for your projects.
08	`get_workspace_board`	Retrieves the ZenHub board view for an entire workspace and repository combination.

See It in Action

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

U Show me the ZenHub board for repository ID '12345678'.

I've retrieved the ZenHub board for repository 12345678. You have 5 pipelines: 'New Issues', 'Backlog', 'In Progress', 'Review/QA', and 'Done'. Would you like to see the issues in a specific pipeline?

U Move issue #45 in repo '12345678' to the 'In Progress' pipeline (ID: '56789') in workspace '98765'.

Successfully moved issue #45 to the 'In Progress' pipeline in your ZenHub workspace.

U What are the estimates for all issues in the current epic?

For the current epic, I found 8 issues. Total estimated points: 34. Current distribution: 3 issues (8 pts), 2 issues (5 pts), 3 issues (3 pts).

Frequently Asked Questions

01 **How do I check the full board status with ZenHub MCP?**

You use `get_repo_board` or `get_workspace_board`. This retrieves all pipelines and issues for a given repository, giving you an instant overview of where everything stands.

02 **Can I change an issue's status using ZenHub MCP?**

Yes. The tool `move_issue_between_pipelines` allows your agent to update the workflow status for any issue, moving it from one pipeline stage to another with a single command.

03 What is required to set story point estimates using ZenHub MCP?

You must use `set_issue_estimate`. You provide the specific issue ID and the desired numerical estimate, and the tool handles updating that field directly in the system.

04 Does ZenHub MCP work with multiple epics?

Yes. You can list all available large features using `list_repo_epics` and then select a specific epic for deeper analysis using `get_epic_data`.

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 `"zenhub": { "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

ZenHub 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 ZenHub. 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	ZenHub MCP
Server ID	`019d7627-7a3a-7168-931c-978c4e4f72a2`
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/zenhub.