MCP SERVER | NO CODE | CLOUD HOSTED

Harness MCP

Manage deployments and audit secrets via chat.

Harness connects your CI/CD and DevOps workflows directly to your AI agent. It lets you manage software delivery pipelines, monitor deployments in real time, and audit platform secrets without opening a complex dashboard.

A+ Quality Score 100/100

ci-cd | pipeline-automation | software-delivery | feature-flags | cloud-cost-management | deployment-monitoring

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

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.

Harness MCP

11 tools available
Cloud-hosted on Vinkius

Your agent gives you full control over your entire software development lifecycle. Instead of navigating multiple dashboards just to check if the latest build passed or to trigger a release, you talk to your AI client. This MCP lets you list all projects across your organization and inspect their pipelines in natural conversation. You can ask for real-time deployment status using `get_execution_status` or even retrieve platform audit logs via `get_audit_logs` to check compliance. It's like having a dedicated DevOps Coordinator available 24/7 right inside your IDE, making it simple to manage everything from infrastructure connectors to secrets.

When you connect this through Vinkius, you get access to all these controls—from listing microservices via `list_services` to triggering an entire release with `execute_pipeline`. It's about getting the results instantly without ever leaving your coding environment.

Core Capabilities

01 — Manage pipelines

List, view details for, and trigger runs of software delivery pipelines.

02 — Monitor deployments

Check the live status and step-by-step progress of any running deployment execution.

03 — Inspect infrastructure assets

View project details, list connected environments, or check available secrets for configuration accuracy.

04 — Audit platform changes

Retrieve historical logs to track who changed what and when within the entire system.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP in Vinkius and provide your Harness API Key, Account ID, and Organization ID.

02 Next, instruct your AI client to perform a specific action, like listing all available projects or checking the status of an active pipeline run.

03 Finally, your agent uses the provided credentials to talk directly to Harness, returning structured data about the requested pipelines, secrets, or executions.

The bottom line is you manage complex DevOps tasks by talking to a single chat interface, letting your AI client do the heavy lifting across multiple systems.

Built For

DevOps Engineers and SREs who are tired of jumping between five different web dashboards just to confirm a deployment status. This MCP gives you full control over the software delivery lifecycle right from your IDE.

DevOps Engineer

They use this to trigger test deployments and check pipeline statuses without leaving their terminal or code editor.

Site Reliability Engineer (SRE)

They monitor execution failures and audit logs immediately, maintaining platform stability during high-stress rollouts.

Release Manager

They oversee multiple projects, list different environments, and ensure all delivery pipelines are configured correctly before a major release.

What Changes When You Connect

01 Real-time visibility: Instead of checking a dashboard, you ask your agent for the status of an execution using `get_execution_status`. You know exactly where the deployment is stuck.

02 Full project oversight: List all organizational projects with `list_projects` and then dive deep into any single pipeline by calling `get_pipeline`. Everything stays in one conversation.

03 Security compliance checks: Need to prove who changed a secret? Use `get_audit_logs` to pull comprehensive change history instantly, simplifying audits.

04 Controlled deployments: When you're ready for production, use `execute_pipeline` to trigger the full run with a simple command. No manual clicks required.

05 Infrastructure mapping: Quickly list all connected environments (`list_environments`) and services (`list_services`), ensuring your next deployment targets the right place.

Real-World Applications

The build failed, but I don't know why.

A developer asks their agent to check the status of the latest run. The agent uses `get_execution_status` and reports that the 'Artifact Push' step is running slowly, pinpointing the exact stage causing the delay.

I need to prove we followed policy on this change.

An auditor asks for a record of who accessed the database credentials last week. The agent executes `get_audit_logs`, providing a clean, chronological list of all access attempts.

I need to roll out a hotfix immediately.

The Release Manager asks their agent to execute the 'Hotfix' pipeline using `execute_pipeline`. The agent confirms the trigger and provides an initial execution ID for tracking.

What services are connected to this project?

A new engineer needs to understand the scope of a microservice. They ask their agent, which uses `list_services` to provide a full inventory of all running components in that project.

Patterns to Avoid

Copy-pasting IDs and endpoints

✗ AVOID

A user manually copies the pipeline ID from one dashboard, pastes it into another tool's URL bar, and then has to find the correct execution status page.

✓ INSTEAD

Just ask your agent directly. Tell it: 'Check the status of the deployment for the E-commerce app.' The agent handles all the ID retrieval using tools like `list_pipelines` and `get_execution_status` automatically.

Forgetting required credentials

✗ AVOID

The user gets an error message saying 'Access Denied' because they didn't enter the correct API Key or Organization ID.

✓ INSTEAD

Remember to subscribe first and provide your Harness API Key, Account ID, and Org ID in Vinkius. This MCP handles authentication for you so you don't have to worry about passing raw credentials.

Treating it like a search engine

✗ AVOID

The user types: 'Show me build status, secrets list, and project name'. The agent gets confused because the requests are unrelated.

✓ INSTEAD

Keep your questions focused. Ask for one thing at a time, like: 'List all projects' (`list_projects`), or 'What is the status of pipeline X?' (`get_execution_status`). Clarity works best.

The Right Fit

Use this MCP if your current process requires coordinating information across multiple distinct systems—specifically, needing to check deployment status, view logs, and manage secrets all from one place. If you only need basic code linting or simple file manipulation, a general coding agent is fine. However, if the task involves anything related to 'who changed what,' 'what's deployed right now,' or 'triggering an entire release cycle,' this MCP is essential. Don't use it if you just want to read documentation; for that, a search tool works better. But if you need to actually *act* on the platform—like triggering `execute_pipeline` or fetching audit logs—this is your go-to resource.

The Pain of Dashboard Overload

Today, checking a single deployment status requires opening Harness, navigating to the correct project, finding the pipeline list, clicking on an execution run, and then looking for the specific step details. This cycle repeats every time you need confirmation.

With this MCP, your AI agent handles all those clicks behind the scenes. You simply ask: 'What's the status of Production Deploy?' And boom—you get the live step-by-step answer immediately. It just works.

Harness for Pipeline and Execution Management

Manual steps like listing all projects, finding a specific secret key, or determining which environments are available used to require multiple distinct API calls and manual cross-referencing. You'd spend half your day just gathering data.

Now, you tell the agent: 'List all projects and their secrets.' It consolidates that information for you instantly. The process is shorter, smarter, and requires zero context switching.

Harness MCP - 11 Tools for DevOps Automation

These tools give you granular control over every part of your CI/CD pipeline, letting your agent manage everything from project listing to running live deployments.

#	TOOL	DESCRIPTION
01	`execute_pipeline`	Starts the process for a specified software pipeline run.
02	`get_audit_logs`	Pulls historical records of all changes made to the platform.
03	`get_execution_status`	Retrieves current status and step details for a specific pipeline run.
04	`get_pipeline`	Gets detailed information, including the YAML definition, for one specific pipeline.
05	`list_connectors`	Displays available infrastructure connections like Git or Kubernetes.
06	`list_environments`	Shows all defined deployment environments within a project, such as staging or production.
07	`list_executions`	Lists multiple completed and active runs for a single pipeline.
08	`list_pipelines`	Retrieves all defined pipelines within a specific project scope.
09	`list_projects`	Lists every organizational project configured in your Harness account.
10	`list_secrets`	Displays the names and types of secrets stored for a given project.
11	`list_services`	Lists all microservices defined within your current DevOps ecosystem.

See It in Action

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

U List all pipelines in project 'E-commerce App'.

I've retrieved the pipelines for 'E-commerce App' (ID: app_502). You have 3 active workflows: 'Production Deploy', 'Staging Sync', and 'Security Scan'. Which one would you like to inspect?

U Execute the 'Production Deploy' pipeline for project ID app_502.

Execution triggered! I've successfully requested the start of 'Production Deploy' (ID: deploy_v1) in project app_502. You can check the status using the `get_execution_status` tool in a few moments.

U Show the status of the latest execution for pipeline deploy_v1.

Retrieving execution status... The latest execution for 'Production Deploy' (ID: exec_992) is currently IN_PROGRESS. 2 out of 5 steps have completed successfully (Build and Unit Tests). Step 3 (Artifact Push) is running. Should I monitor this for you?

Frequently Asked Questions

01 **How do I find my Harness API Key?**

Log in to Harness, go to **My Profile** (bottom left), and select **API Keys**. You can generate either a Personal API Key or use a Service Account Token from there.

02 **Where can I find my Account ID and Org ID?**

The Account ID is visible in your browser's URL when logged in (e.g., `account/XYZ123`). The Organization ID is typically found in the project settings or sidebar (default is often 'default').

03 **Can I trigger a pipeline execution via this integration?**

Yes! Use the `execute_pipeline` tool by providing the project ID and pipeline ID. Harness will initiate the workflow immediately.

04 **Is the integration secure for managing secrets?**

Absolutely. The integration only lists the metadata of your secrets (like names and identifiers) through your API key. Your credentials are encrypted and stored securely in the Vinkius Cloud.

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 `"harness": { "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

Harness 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 Harness. 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	Harness MCP
Server ID	`019d75ae-615f-715e-a828-fc152b3d63c5`
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/harness.