MCP SERVER | NO CODE | CLOUD HOSTED

Concord MCP for AI Agents

Manage CI/CD Workflows and Deployment Logs from Chat

Concord provides your AI agent with full programmatic access to your self-hosted CI/CD workflow orchestration platform. It lets you manage organizations, view project structures, run new deployments, track active processes, and pull detailed execution logs—all from natural language commands.

A+ Quality Score 98.33/100

workflow-orchestration | ci-cd | process-automation | log-analysis | pipeline-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/concord-workflow-orchestration

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.

Concord (Workflow Orchestration) MCP

10 tools available
Cloud-hosted on Vinkius

Managing complex workflows usually means jumping between dashboards: checking the status dashboard, pulling logs into a text editor, cross-referencing project details in another tab. This MCP changes that. It connects your agent directly to your Concord instance, giving it visibility into every part of your CI/CD process.

You can ask your AI client to list all organizations and projects across your entire setup. Need to check a failing deployment? Simply ask the agent for the logs for a specific run, and it retrieves the failure context instantly. You don't have to manually copy IDs or navigate deep into menus; you just describe what you need—a running process status, project structure, or log output.

Whether you are troubleshooting an issue during an incident response or simply auditing your system's current state, this MCP centralizes that operational knowledge. When connected through Vinkius, it makes Concord a natural part of your existing AI toolset, allowing your agent to handle complex process management tasks without needing dedicated UI interaction.

Core Capabilities

01 — Discovering Project Scope

Retrieve the full organizational structure, listing all configured organizations and their contained projects.

02 — Tracking Workflow Status

Get a live overview of every currently running workflow process and retrieve detailed metrics for any specific instance.

03 — Controlling Processes

Trigger new deployments or manually stop runaway processes using conversational commands.

04 — Diagnosing Failures

Pull the complete, historical execution logs for any process instance to find the exact failure context.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/concord-workflow-orchestration** — connect your AI agent in three steps.

01 Add the Concord integration details—your instance URL and API Token—to your AI client's toolset.

02 Your agent uses the stored credentials to connect directly to your private CI/CD workflow platform.

03 You issue a natural language command (e.g., 'Show me the logs for project X'), and the agent executes the necessary action via Concord, returning the structured data or text output.

The bottom line is that you manage your entire CI/CD pipeline conversationally, without leaving your chat interface.

Built For

This MCP is essential for DevOps Engineers and Platform Teams. If you spend too much time switching between dashboards (the status view, the log viewer, the project selector) just to diagnose a deployment failure, this tool gives your agent the single pane of glass you need. It lets you act on complex systems using only plain language.

DevOps Engineer

Diagnosing failed pipelines by requesting specific execution logs or listing all currently running processes during an incident.

Platform Team Lead

Auditing the entire infrastructure to list and categorize every organization, project, and repository configured in Concord.

Release Manager

Monitoring live deployments, identifying stale or stuck workflows using list_running_processes, and terminating them directly from chat.

What Changes When You Connect

01 Stop context switching. Instead of opening the dashboard, navigating to 'Processes,' finding the ID, and then pulling logs into a separate window, you simply ask your agent for the details using the get_process_log tool.

02 Audit quickly across large systems. Need to know what departments exist? Use list_organizations to pull an inventory list of every single organizational silo in minutes.

03 Handle incidents without hands-on access. If a deployment is stuck, you can use list_running_processes to identify the runaway job and then terminate_process to shut it down immediately.

04 Understand project dependencies instantly. By calling list_projects or get_project_details, your agent provides a clear map of which repositories are linked to which projects.

05 Automate execution triggers. Instead of manually clicking 'Run' on a build job, you can use start_process and let your agent handle the workflow initiation step.

Real-World Applications

Investigating an Overnight Failure

A Release Manager notices that staging deployment failed. Instead of hunting through logs manually, they ask their agent to pull the logs for the specific process run ID, immediately identifying the database connection error and narrowing down the fix.

System Audit Before Expansion

A Platform Team needs to know how many business units are running workflows. They use list_organizations to pull a definitive count of all existing organizations, which helps them scope out required resource allocations for expansion.

Stopping a Bad Deployment

During testing, an engineer realizes a process is running with incorrect credentials and could cause damage. They immediately tell their agent to list_running_processes to find the job ID and then terminate_process before it finishes.

Mapping Project Scope

A new team member needs to understand all deployment targets. They ask their agent to get_project_details for a main product line, which returns not just the project name but also every associated repository and its purpose.

Patterns to Avoid

Trying to manually track process history

✗ AVOID

Manually clicking through dozens of completed jobs in the UI just to find a specific deployment status from last week. This is slow, tedious, and prone to error.

✓ INSTEAD

Tell your agent to list_processes and filter by date range or job name. It compiles the history into an easily readable format instantly.

Forgetting which projects exist

✗ AVOID

Assuming a new team has connected all necessary repositories because they are 'under the project.' You might miss crucial dependencies.

✓ INSTEAD

Use list_repositories in conjunction with get_project_details. This ensures your agent checks every linked repository to confirm full coverage.

Confusing process status

✗ AVOID

Seeing a job listed as 'running' but not knowing if it's stalled, healthy, or just starting up.

✓ INSTEAD

Use get_process. This tool provides the detailed metadata—the true state and current task—so your agent can tell you exactly what that process is doing right now.

The Right Fit

Use this MCP if your primary pain point is context switching during CI/CD diagnostics. If you spend more than five minutes per incident trying to piece together logs, statuses, and project structures from different dashboards, this tool solves that by making Concord data conversational.

However, don't use it if all you need is a simple list of names—for instance, if you only want to see the current user roster. For basic

inventory tasks unrelated to workflow runs (like managing users), an API endpoint for directory services would be better suited. If your process failures are always due to bad code and never environmental setup, you might just need a static logging service, not full orchestration control.

Concord MCP: Managing Workflow Orchestration Status

Right now, diagnosing a failed deployment requires a painful sequence of clicks. You check the main dashboard to see if something is running. Then you copy the process ID and paste it into another tab to get details. If that fails, you have to find the log viewer, search for the run ID again, and finally manually scroll through hundreds of lines of text until you spot 'ERROR.'

With this MCP, your agent handles the whole sequence in one go. You just ask: 'What happened with deployment X?' The agent coordinates finding the process details using get_process, pulling the historical logs via get_process_log, and summarizing the failure context for you. It's all conversational.

Concord MCP: Auditing Project Structure and Dependencies

Before this integration, understanding your full project landscape was a manual scavenger hunt. You'd have to ask three different people—DevOps for the projects, Architecture for the organizations, and Security for the repositories—just to map out what services existed.

Now, you can use list_organizations followed by list_projects and then get_project_details in sequence. Your agent builds a comprehensive, real-time inventory of your entire deployment surface area without anyone leaving the chat window.

Concord (Workflow Orchestration) MCP: 10 Process Management Tools

Use these tools to manage the entire lifecycle of your CI/CD processes, from listing organizations to terminating active runs.

#	TOOL	DESCRIPTION
01	`list_projects`	Gets a list of all projects within a specified organization.
02	`list_repositories`	Retrieves a list of repositories that are configured for a given project.
03	`list_running_processes`	Quickly lists all processes that are currently running in the workflow system.
04	`start_process`	Triggers a new workflow execution run for a defined project or template.
05	`terminate_process`	Stops an active workflow process execution immediately.
06	`get_process`	Retrieves detailed status and metadata about a single, specific process run.
07	`get_process_log`	Fetches the full text logs for any given workflow execution instance.
08	`get_project_details`	Retrieves comprehensive details about a specific project within Concord.
09	`list_organizations`	Gets an inventory list of every organization configured in your system.
10	`list_processes`	Retrieves a full history listing of all past and present process executions.

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 active deployments that are currently running or have failed in the last 24 hours.

Here's a summary of recent activity:

- **Project Phoenix:** Running (Started 5 minutes ago). Status: In Progress. Last task: Build container image.
- **Billing API:** Failed (Completed 3 hours ago). Error context: Connection Timeout at step 'Validate credentials'.
- **Frontend UI:** Successful (Completed 1 day ago). Final status: Deployed to Staging.

U I need the full logs for the Billing API failure. What was wrong?

Pulling detailed execution logs now...

[Log Snippet Start]

`Task 'Validate credentials' failed: ECONNREFUSED 127.0.0.1:5432`. The service appears to be inaccessible or the port is closed.

[Log Snippet End]

The agent suggests checking network connectivity first.

U List all organizations and their projects so I can plan the next audit.

Your Concord instance manages 4 distinct organizational units:

1. **Engineering:** 24 active projects.
2. **Data Platform:** 8 active projects (Focus: ETL workflows).
3. **QA Automation:** 15 active projects.
4. **Finance Systems:** 3 active projects.

Which organization do you want to investigate next?

Frequently Asked Questions

01 **How does Concord MCP help me track deployments?**

The agent provides a unified view of your entire CI/CD lifecycle. You can list all processes, see which ones are running right now, and get status updates on past runs without switching tabs or dashboards.

02 **Can I use Concord MCP to find out what repositories exist?**

Yes. If you tell the agent a specific project name, it can retrieve all associated repositories for that project. This is useful for auditing dependencies and understanding your code base scope.

03 **What if my deployment fails? Can Concord MCP help me debug?**

Absolutely. You ask the agent to pull the logs for a specific process run, and it retrieves the complete text output, letting you see the exact error context—like which connection failed or what line of code broke.

04 **Does Concord MCP only work with my self-hosted instance?**

Yes. This MCP is designed to connect specifically to your own private, self-hosted Concord deployment. It keeps all your workflow data securely within your environment.

05 **How do I start a new process using the Concord MCP?**

You simply ask your agent to trigger a run for a specific project. The agent handles the necessary API calls, starts the deployment sequence in Concord, and reports back when it begins.

06 **Is listing organizations part of the Concord MCP?**

Yes, this is one of its core capabilities. It gives you a high-level inventory of every organizational silo within your system, which is critical for large platform audits and scoping new work.

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 `"concord-workflow-orchestration": { "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

Concord (Workflow Orchestration) 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 Concord (Workflow Orchestration). 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	Concord (Workflow Orchestration) MCP
Server ID	`019d7579-5d2d-7083-9386-4917c46b0e4c`
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/concord-workflow-orchestration.