MCP SERVER | NO CODE | CLOUD HOSTED

# Codefresh MCP for AI Agents

## Manage CI/CD Pipelines and Kubernetes Cluster Deployments

Connect Codefresh to your AI client to manage CI/CD and GitOps workflows. This MCP lets you list pipelines, trigger builds, monitor Kubernetes clusters, and audit environment secrets—all through natural conversation. You get full visibility into software deployment status without opening a dashboard.

F Quality Score 3.6/100

kubernetes | continuous-delivery | pipeline-automation | gitops | container-orchestration | deployment-management

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

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.

# Codefresh MCP

8 tools available
Cloud-hosted on Vinkius

Codefresh gives your AI agent direct access to your entire continuous delivery infrastructure. Instead of jumping between dashboards or writing complex API calls, you just talk to your client. Your agent can list every pipeline in the account, check the status of recent builds, and even kick off new deployments for specific branches. It monitors all connected Kubernetes clusters so you know exactly where your code is running. The whole thing works naturally; whether you're checking a secret variable or verifying cluster connectivity, it handles it. Connecting this MCP to Vinkius means you get access right alongside hundreds of other tools, keeping your workflow consolidated and hands-free.

## Core Capabilities

**01 — Check pipeline status**

List all CI/CD pipelines in the account or retrieve detailed information for a specific one.

**02 — Monitor recent builds**

View execution details and overall status for multiple recent build workflows.

**03 — Trigger new deployments**

Start a fresh build run on any specified pipeline, including defining target branches or variables.

**04 — Audit environments**

List all shared contexts, secrets, and environment variables used across your workflows to verify security settings.

**05 — Verify cluster connectivity**

Get a list of all connected Kubernetes and delivery clusters so you know where deployments are targeting.

# One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this Codefresh MCP on Vinkius.

02 Enter your unique Codefresh API Key into the connection settings (find it in User Settings > API Keys).

03 Ask your AI client to perform a task, like 'Check the status of the main app build,' and the agent executes the command.

The bottom line is that you connect once, and your AI client can manage all your complex CI/CD tasks using simple chat prompts.

## Built For

DevOps Engineers who spend too much time clicking through dashboards at 2 AM. Release Managers needing quick proof of deployment success rates without opening the Codefresh UI. Software Developers who just want to verify build logs or environmental contexts straight from their chat window.

**DevOps Engineer**

Uses this MCP to monitor pipeline health, list configurations, and trigger manual builds using natural language commands.

**Release Manager**

Audits deployment success rates and verifies the status of multiple connected Kubernetes clusters instantly from their agent interface.

**Software Developer**

Quickly looks up build execution details or checks shared environment variables to confirm development context without leaving their primary workflow.

## What Changes When You Connect

01 Stop switching tabs. You can list all pipelines or check build status, using the `list_codefresh_pipelines` tool directly in your chat interface.

02 Verify environment variables instantly. Use `list_shared_contexts` to audit secrets and shared variables without navigating complex settings pages.

03 Control deployments from anywhere. With `trigger_codefresh_build`, you can kick off a new build for any pipeline, specifying the exact branch or variable needed.

04 Get full visibility into delivery targets using `list_delivery_clusters`. Know exactly which Kubernetes clusters are connected and ready for deployment.

05 Deep dive into history. Use `get_build_execution_details` to pull up the full status and logs for any build workflow, quickly diagnosing failures.

## Real-World Applications

**Need to check if the new API service deployment went out correctly?**

Instead of opening the dashboard, ask your agent to use `list_delivery_clusters` to confirm the target cluster name. Then, use `get_build_execution_details` on the latest build ID to verify the successful rollout.

**The QA team needs a fresh test environment built.**

Ask your agent to run `list_codefresh_pipelines` to find the 'qa-deployment' pipeline. Then, instruct it to use `trigger_codefresh_build`, specifying the 'develop' branch and setting necessary variables.

**A developer needs to check what secrets are available for a specific service.**

The agent uses `list_shared_contexts` to display all relevant environment variables, allowing the developer to confirm credentials before starting local testing. This saves minutes of manual auditing.

**I need to see if any pipelines are configured for a specific microservice.**

The agent runs `list_codefresh_pipelines` and filters by service name. You immediately get the pipeline configuration details without browsing through hundreds of entries in the web UI.

# Patterns to Avoid

### Manual dashboard navigation

✗ AVOID

Logging into Codefresh, navigating to Pipelines, clicking a build ID, then scrolling down to find an expired secret variable.

✓ INSTEAD

Ask your agent to run `list_shared_contexts` directly. It pulls the secret details and context variables you need without any clicks or logins.

### Guessing which pipeline needs triggering

✗ AVOID

Remembering that the 'backend' service uses a different build flow than the 'frontend', leading to running the wrong `trigger_codefresh_build` command.

✓ INSTEAD

First, run `list_codefresh_pipelines` to see all available pipelines. Then, provide the exact name and parameters when you ask your agent to trigger the build.

### Forgetting which clusters are connected

✗ AVOID

Thinking a new staging cluster was added but not knowing its official name or if it's ready for deployment.

✓ INSTEAD

Ask your agent to run `list_delivery_clusters`. It provides an immediate, comprehensive list of every connected Kubernetes target.

## The Right Fit

Use this Codefresh MCP if you need operational visibility into continuous delivery from a chat interface. Specifically, use it when you need to audit secrets ( `list_shared_contexts` ), manage build triggers ( `trigger_codefresh_build` ), or monitor cluster health across multiple environments (using `list_delivery_clusters` ). Don't use this if your only goal is writing pipeline YAML files; that requires the Codefresh UI. If you just want a high-level overview of *all* pipelines, run `list_codefresh_pipelines` . But if you need deep technical details on one build, start with `get_build_execution_details` .

# Codefresh MCP for AI Agents: Streamlining CI/CD Pipeline Oversight

Right now, managing software deployments means juggling multiple dashboards. You have to check the build status on one tab, verify environment variables in another, and then jump over to a separate cluster view just to confirm everything is pointing correctly. It's slow, it’s prone to human error, and frankly, it wastes time.

With this MCP, your agent handles the whole flow. You can ask for a full list of pipelines ( `list_codefresh_pipelines` ) or check specific build status details using `get_build_execution_details` . The result is immediate answers about your entire delivery graph.

# Codefresh MCP for AI Agents: Auditing GitOps and Cluster Context

Manually verifying deployment targets or auditing secrets requires opening the cluster management view, remembering which variables are shared, and cross-referencing them with your build logs. It's a multi-step process that forces you to context-switch constantly.

Now, you just ask for it. You can run `list_delivery_clusters` to confirm all connected targets or use `list_shared_contexts` to pull up every necessary secret and variable in one go. Your agent makes the entire system transparent.

# 8 Tools in the Codefresh MCP for Pipeline Management

Use these tools to list configurations, trigger builds, check cluster health, and retrieve detailed logs across all your CI/CD workflows.

| # | TOOL | DESCRIPTION |
|---|---|---|
| 01 | `get_build_execution_details` | Fetches the detailed status and full execution history for a single, specific build run. |
| 02 | `get_my_codefresh_profile` | Retrieves core information about the authenticated user and the connected Codefresh account details. |
| 03 | `get_pipeline_configuration` | Gets detailed settings and metadata for a single, specified CI/CD pipeline. |
| 04 | `list_codefresh_builds` | Lists all recent build workflows that have run in the account history. |
| 05 | `list_delivery_clusters` | Provides a list of every connected Kubernetes and delivery cluster monitored by Codefresh. |
| 06 | `list_shared_contexts` | Lists all shared environment contexts, including sensitive secrets and variables used across pipelines. |
| 07 | `list_codefresh_pipelines` | Retrieves a list of every defined CI/CD pipeline available in the account. |
| 08 | `trigger_codefresh_build` | Starts and initiates a brand new build for a specified pipeline, allowing you to set branches or variables. |

# See It in Action

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

**U** What's the status of the latest builds for my main application?

**Recent Build Status**

- `main-app-deploy` : **SUCCESS**. Deployed to `staging-us-east` . (ID: 605c...123)
- `api-service-ci` : **FAILED**. Error on step 'Test Coverage'. Check logs for details.
- `frontend-build` : **RUNNING**. Currently deploying to `staging-us-west` .

You can ask me to get the detailed logs for the failed API build.

**U** List all available pipelines and tell me which one manages user auth.

**Codefresh Pipelines Found**

1. `main-app-deploy` : Core service deployment (Primary)
2. `api-service-ci` : API logic build/test (Secondary)
3. `user-auth-pipeline` : **Handles user authentication flow.**
4. `frontend-build` : UI component builds.

The `user-auth-pipeline` seems like the right one for what you need.

U I need to run a new build on the payment service, using variables X and Y.

Initiating build for 'payment-service'...

✅ **Build Started Successfully.**

- **Pipeline:** `payment-service-ci`
- **Branch:** `develop`
- **Variables Set:** `X`, `Y`
- **New Build ID:** 605d...456

I've started tracking it. I'll let you know if the status changes.

# Frequently Asked Questions

01 **How does Codefresh MCP help me monitor my CI/CD pipelines?**

This MCP lets your agent list all existing pipelines and gives you real-time visibility into their configuration. You can check the health of any pipeline or get detailed status updates on recent builds without leaving your chat interface.

02 **Can I use Codefresh MCP to trigger a manual build?**

Yes, you can start new builds using this MCP. Simply tell your agent which pipeline name and branch you want, and it initiates the deployment process for you immediately.

03 **What if I need to check environment variables or secrets?**

You can audit all shared contexts like secrets and variables using this MCP. It lists everything used across your workflows in one place, so you always know what data is accessible during deployment.

04 **Does Codefresh MCP help with Kubernetes cluster status?**

Absolutely. This MCP allows you to list all connected delivery clusters and check the current build execution details for deployments targeting those specific environments.

05 **Is this better than using the Codefresh web dashboard?**

It's faster because it eliminates clicks. Instead of navigating deep into dashboards, you ask your agent a question and get an immediate, concise answer or action taken directly in the chat.

# 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 `"codefresh": { "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

# Codefresh 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 Codefresh. 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 | Codefresh MCP |
| Server ID | `019d7576-69f4-71dc-822e-6c642638e28e` |
| 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/codefresh.