MCP SERVER | NO CODE | CLOUD HOSTED

Northflank MCP

Orchestrate Cloud Deployment from Conversation

Northflank MCP lets you manage your entire developer cloud infrastructure directly through conversation. Orchestrate microservices, deploy new code builds, audit complex background jobs, and provision or delete entire project ecosystems without touching a dashboard or running a single CLI command.

A+ Quality Score 100/100

microservices | cloud-deployment | ci-cd | infrastructure-orchestration | backend-hosting

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/northflank-developer-cloud-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.

Northflank (Developer Cloud & Orchestration) MCP

10 tools available
Cloud-hosted on Vinkius

Control your enterprise development platform using natural language. This connection gives your agent full oversight of your microservice orchestration and continuous deployment pipelines across multiple cloud regions. You can list every managed project and get detailed metadata, checking things like which geographic data centers (AWS, GCP, Azure) are connected. Need to restart a service? Simply ask the agent to cycle container replicas for specific applications. The MCP also handles operational tasks like listing isolated batch jobs or auditing secret groups—verifying environment variables across different virtual private cloud boundaries. When you connect this via Vinkius, your AI client treats Northflank as just another tool in its belt, letting you manage complex infrastructure from anywhere.

Core Capabilities

01 — Manage Project Lifecycle

Provision new isolated project spaces or permanently tear down existing microservice ecosystems.

02 — Audit Service Health

Retrieve the precise resource allocation and structural details for any running application service.

03 — Control Deployments

Manually trigger fresh continuous integration builds to update production assets or verify recent code merges.

04 — Restart Services

Gracefully cycle container replicas for a specific service, clearing accumulated transient memory and restoring normal performance timing.

05 — Monitor Background Processes

List and inspect isolated batch or cron jobs to track periodic tasks like heavy database aggregations.

06 — Verify Credentials

Access metadata for logical secret groups, confirming environment variable mappings across your cloud boundaries.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/northflank-developer-cloud-orchestration** — connect your AI agent in three steps.

01 Subscribe to this MCP on Vinkius and provide the required Northflank API Token.

02 Connect your preferred AI client (like Cursor or Claude) to the catalog, granting it access to the tool.

03 Ask your agent a natural language question—for example, 'List all active microservices in Project X'—and let it perform the necessary cloud actions.

The bottom line is you can use natural conversation to execute complex DevOps operations that used to require multiple specialized dashboards and terminal commands.

Built For

This MCP is for the Platform Engineer who gets frustrated having to jump between a dozen different cloud provider consoles just to verify if a deployment worked. It's also for the Senior Backend Developer who needs quick, accurate data on resource limits without bothering an Ops team.

DevOps Engineer

Uses this MCP to monitor microservice health across multiple regions and trigger production deployments using simple conversation prompts.

Backend Developer

Verifies service resource allocations, checks the status of scheduled background jobs, and audits secret vault mappings directly from their development terminal.

Platform Team Lead

Manages organizational project boundaries, tears down old microservice ecosystems, and verifies secure credential mappings across different environments efficiently.

What Changes When You Connect

01 Instant Service Audit: Instead of manually checking dashboards, ask your agent to use `get_service` to instantly verify a service's structural anatomy and current resource boundaries.

02 Rapid Deployments: When code changes, you don't navigate to the CI/CD dashboard. Just prompt your agent to `trigger_build`, starting the deployment immediately.

03 Clean Up Effortlessly: Need to decommission an old testing environment? Use `delete_project` to permanently tear down a project and all its microservices in one command.

04 Proactive Maintenance: Don't wait for failure. Tell your agent to use `restart_service` on critical applications to cycle replicas and clear memory buildup proactively.

05 Security Visibility: Quickly verify credentials by asking the agent to use `list_secrets`, checking environment variable mappings across all connected VPC boundaries.

06 Full Oversight: Need a quick inventory? Use `list_projects` or `list_services` to get an immediate, comprehensive list of every resource in your account.

Real-World Applications

Investigating Production Outages

A payment service is running slow. Instead of logging into three different consoles, the engineer asks their agent to `get_service` for that specific application. The agent returns immediate data on CPU throttling and RAM allocation boundaries, pointing directly to the resource bottleneck.

Preparing a New Team Environment

A new team needs a sandbox environment. Instead of manually setting up networking rules, the platform lead asks the agent to `create_project`. The system provisions an isolated project space instantly, ready for development.

Auditing Security Changes

Before migrating data, the security team needs to confirm where database connection strings are used. They ask the agent to use `list_secrets` and get a map of all environment variable types across different operational zones.

Verifying Scheduled Tasks

A nightly report is failing, but nobody knows which cron job handles it. The developer asks the agent to `list_jobs`, retrieving a list of isolated batch processes and confirming the exact scheduled task that needs fixing.

Patterns to Avoid

Over-reliance on Manual Dashboards

✗ AVOID

Opening 12 different tabs, navigating to 'Services', then clicking through filters to find a single microservice's resource allocation.

✓ INSTEAD

Ask your agent to `get_service` directly. It retrieves the exact structural anatomy and necessary metadata in one step.

Misunderstanding Project Scope

✗ AVOID

Trying to delete a service without first confirming if it's connected to vital secrets or background jobs.

✓ INSTEAD

First, use `list_secrets` to check the dependencies, then confirm with `get_project` before running `delete_project`. Always check scope.

Ignoring Deployment Triggers

✗ AVOID

Assuming that simply pushing code to GitHub automatically deploys it to production and forgetting to restart containers.

✓ INSTEAD

After merging, you must explicitly tell your agent to `trigger_build` first, then use `restart_service` on the target application.

The Right Fit

Use this MCP if your job involves orchestrating complex, multi-stage deployments across several cloud boundaries. Specifically, if you need to check resource allocations (`get_service`), manage entire project lifecycles (`create_project` / `delete_project`), or audit credentials and jobs (`list_secrets` / `list_jobs`). Don't use this if you only need to read simple data points; for instance, if you just want a list of all user names without knowing their service status, that’s too narrow. This MCP is for the full lifecycle control panel—the place where infrastructure orchestration happens.

The Pain of Jumping Between Dashboards

Today, changing a simple microservice configuration means logging into the Northflank console. You navigate to the Project Overview, click into Services, then find the specific instance you need. If you want to check its resource limits or see if it's connected to a secret vault, you usually have to run two or three different reports and copy-paste the data yourself.

With this MCP, your agent handles that whole sequence in one chat window. You don't navigate anywhere; you just ask for the information—like checking its resource allocation—and get the clean, actionable answer immediately.

Northflank (Developer Cloud & Orchestration) MCP

The most time sink is managing the full deployment lifecycle. You have to remember to check if a project needs resources provisioned, list all dependent services, and then manually run the build command before you even consider restarting anything.

Now, you simply tell your agent what you want done—for example, 'Deploy the latest version of payment processor.' The MCP sequences `list_services`, calls `trigger_build`, and finally runs `restart_service` automatically. It just works.

Northflank (Developer Cloud & Orchestration) - 10 Tools

Use these tools to programmatically list projects, deploy microservices, check resource metadata, and audit background jobs across your developer cloud environment.

#	TOOL	DESCRIPTION
01	create_project	Creates a completely new, isolated Northflank Project space for development work.
02	delete_project	Permanently removes an entire project boundary and all associated microservices and running jobs.
03	get_project	Retrieves detailed regional metadata about a specific Northflank Project, including its cloud boundaries.
04	get_service	Fetches the exact structural anatomy and scaling needs of a single running application service.
05	list_jobs	Lists all isolated batch and cron jobs that run within a specific project.
06	list_projects	Fetches the mandatory list of top-level organizational Project IDs.
07	list_secrets	Lists all secret group dictionaries, showing environment variables mapped into running services or jobs.
08	list_services	Retrieves a list of explicit application or microservice instances inside a project.
09	restart_service	Gracefully cycles the container replicas for a specific service to clear transient memory buildup.
10	trigger_build	Sends a command to Northflank builders to compile and deploy the latest source code.

See It in Action

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

U List all services in the 'api-gateway' project

I've retrieved the services for 'api-gateway' (ID: gate-123). Highlights include 'Auth-Service' (Status: Running), 'Payment-Processor' (Status: Running), and 'Notification-Node'. Would you like to check the resource allocation for the auth service?

U Trigger a fresh build for service 'gate-123' in project 'proj-abc'

Triggering production build... Success. I've signaled Northflank to pull the latest code from GitHub and start the build cycle for 'Auth-Service'. I'll monitor the progress and let you know once the new image is live.

U Show me the secret groups configured in my project

Retrieving secret vaults... I've identified 2 logical groups: 'Production-Secrets' (Group ID: sec-987) and 'Shared-API-Keys'. I can provide the names of the environment variables mapped within these groups if you'd like.

Frequently Asked Questions

01 **How do I list all services using Northflank (Developer Cloud & Orchestration) MCP?**

You use the `list_services` tool. This fetches a complete roster of every explicit application or microservice instance running inside your current project.

02 **What is the difference between `get_project` and `list_projects` in Northflank (Developer Cloud & Orchestration) MCP?**

`list_projects` gives you a list of all top-level organizational Project IDs. `get_project` allows you to dive deep into one specific project ID to retrieve its detailed regional metadata.

03 **Can I restart my service using Northflank (Developer Cloud & Orchestration) MCP?**

Yes, you use the `restart_service` tool. This command gracefully cycles container replicas for a specific application, clearing out any transient memory buildup.

04 **I need to find all my secrets, which tool should I use in Northflank (Developer Cloud & Orchestration) MCP?**

Use `list_secrets`. This lists all secret group dictionaries and shows you exactly which environment variables are mapped into your running services or jobs.

05 **What if I want to tear down an entire project? Which tool handles that in Northflank (Developer Cloud & Orchestration) MCP?**

Use the `delete_project` tool. This permanently removes a Project and all its cascading microservices, effectively cleaning up the entire ecosystem.

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 `"northflank-developer-cloud-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

Northflank (Developer Cloud & 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 Northflank (Developer Cloud & 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	Northflank (Developer Cloud & Orchestration) MCP
Server ID	`019d75df-75ba-730e-afbe-532d05469f81`
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/northflank-developer-cloud-orchestration.