MCP SERVER | NO CODE | CLOUD HOSTED

Fly.io Alternative MCP

Manage VMs and Volumes via Conversation

Fly.io Alternative MCP gives your agent full command over edge compute infrastructure. List apps, create machines in any global region, manage persistent storage volumes, and audit SSL certificates using natural conversation.

A+ Quality Score 98.33/100

edge-computing | virtual-machines | server-management | deployment | infrastructure-as-code | cloud-hosting

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/flyio-alternative

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.

Fly.io MCP

13 tools available
Cloud-hosted on Vinkius

Managing distributed applications used to mean jumping between dashboards or running complex CLI commands across multiple regions. Now, you can talk to your agent and have it handle the whole process for you. This MCP connects your account directly to the core controls of your edge infrastructure. Need a new machine in Sydney? Want to check if a volume survived a redeployment? Just ask. Your agent handles the complexity—it's like having an embedded, expert DevOps engineer who lives inside your chat window. It manages everything from provisioning machines with specific CPU and memory limits to ensuring persistent storage volumes are attached correctly. When you connect this MCP via Vinkius, you get instant access to all these controls without ever touching a dashboard again.

Core Capabilities

01 — View Application Overview

List all deployed applications and check their status across the global network.

02 — Manage Machine Lifecycle

Create, start, stop, restart, update, or delete individual virtual machines (VMs) with fine-grained control over resources and regions.

03 — Handle Persistent Storage

List and create volumes that keep data safe even if the attached machine is deleted or restarted.

04 — Audit Security Certificates

Review TLS/SSL certificate status for your app's domains to ensure secure connectivity.

05 — Deploy Globally

Provision and manage compute resources across multiple worldwide regions (like iad, fra, or syd) for low latency access.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Fly.io Personal Access Token.

02 Your agent uses the token to authenticate with your edge compute account.

03 You speak naturally—for example, 'Create a machine in NRT'—and your agent executes the necessary API calls.

The bottom line is that you stop using CLIs or web dashboards and start giving direct instructions to your AI client.

Built For

This MCP is for Platform Engineers and DevOps Teams who are tired of context-switching between dashboards just to manage a few VMs. If you spend more time clicking 'Check Status' than actually building, this is for you.

Platform Engineer

Needs to provision and scale compute resources across diverse global regions without having to manually SSH into every VM.

DevOps Team Lead

Must inspect machine configurations, verify persistent storage attachments, and audit certificate status quickly after a deployment failure.

Full-Stack Developer

Needs to check the live status of machines or restart a problematic VM directly from their chat interface instead of digging through logs.

What Changes When You Connect

01 You gain immediate control over the entire machine lifecycle. Need to stop a running VM or restart one that's misbehaving? Use `stop_machine` or `restart_machine` instead of logging into the dashboard.

02 Managing data persistence becomes simple with tools like `create_volume` and `list_volumes`. You can guarantee your state survives machine restarts without manual setup.

03 Global deployment is streamlined. Whether you need compute in Frankfurt or Sydney, you control it all by using `create_machine`, specifying the exact region needed for low latency.

04 Security auditing gets faster. Instead of hunting through certificates, simply run `list_certificates` to see if your app's domains are issued and healthy.

05 You get a single source of truth. Use `list_apps` first to map out all services, then use `get_machine` on specific apps to drill down into details.

Real-World Applications

The database needs more disk space.

A developer realizes the main app's data volume is filling up. Instead of submitting a ticket, they ask their agent to use `create_volume` and allocate 20GB, then instruct it on how to attach that new storage unit.

The web service keeps crashing at peak load.

An operations engineer notices erratic CPU usage. They ask the agent to use `list_machines` to identify all running instances, and then issue a command using `update_machine` to increase memory allocation across the board.

We need to launch a new regional endpoint.

A product manager dictates that the app needs better performance in Brazil. The engineer uses their agent to execute `create_machine`, specifying the desired Brazilian region and required resource limits immediately.

The service was deployed, but HTTPS fails.

After a deployment, connectivity tests fail. They use the MCP to run `list_certificates` right away, confirming if the certificate is 'pending' or if it needs manual re-issuance.

Patterns to Avoid

Running repetitive checks

✗ AVOID

Manually checking three separate dashboards (App Status, Volumes, Certificates) just to confirm the health of a single service.

✓ INSTEAD

Use the MCP to first run `list_apps`, then `get_machine` for the specific app ID, and finally `list_certificates`. This groups all necessary status checks into one conversation flow.

Ignoring resource limits

✗ AVOID

Creating a machine using default settings that are too small, leading to crashes when load increases.

✓ INSTEAD

Always check the `get_machine` details first. When creating resources with `create_machine`, specify explicit CPU cores and memory in MB to match expected peak usage.

Forgetting data persistence

✗ AVOID

Deleting a machine because it's 'obsolete,' only to realize the critical database logs were stored on that machine.

✓ INSTEAD

Before any deletion, run `list_volumes` and confirm the required persistent storage is listed. If necessary, use `create_volume` first.

The Right Fit

Use this MCP if your primary workflow involves managing the operational state of deployed compute units (VMs) across multiple regions. You need to automate machine scaling, volume attachment, and certificate auditing. Don't use it if you are building application logic or writing code; that requires a general code completion tool. If you only ever manage simple data records without needing VM lifecycle control, look for a generic database MCP instead. But if your pain point is 'I have to click three different dashboards to know if my service is running and connected,' then this MCP gives you the direct conversational access you need.

The Dashboard Fatigue

Today, checking on your edge infrastructure means logging into App Status, clicking over to Volumes Management, and finally hopping to Security Certificates. You copy IDs here, paste them there, wait for a page refresh, and then repeat the entire miserable cycle just to get a single picture of health.

With this MCP, you simply ask your agent what's wrong. It gathers all that information—machine status, volume attachment details, and SSL certificate validity—and presents it back to you in one conversational response. You get answers without the clicks.

Fly.io Alternative MCP: Full Control Over Your Edge Compute

The manual steps that disappear are the context-switching, the ID hunting, and the 'Did I remember to check Volumes?' anxiety. No more guessing if a machine has its required persistent data attached.

You manage your entire global infrastructure—from deployment to storage cleanup—via natural language prompts. You're not just reading status; you're commanding changes.

Fly.io Alternative MCP: 13 Tools

These tools allow you to analyze, manage, and directly control every aspect of your virtual machine fleet, persistent storage, and network security settings.

#	TOOL	DESCRIPTION
01	`create_machine`	Builds and deploys a new virtual machine in your app, allowing you to specify the Docker image, region, CPU cores, and memory size.
02	`create_volume`	Sets up persistent storage volume for your app, specifying the name and total size in gigabytes.
03	`delete_machine`	Permanently removes a virtual machine; this action cannot be undone.
04	`get_app`	Retrieves the name, status, and network details for a specific application within your account.
05	`get_machine`	Fetches all configuration details and current state for a specified machine.
06	`list_apps`	Retrieves a comprehensive list of every application you have deployed across your global edge network.
07	`list_certificates`	Lists all TLS/SSL certificates for an app, showing their hostname and current issuance status.
08	`list_machines`	Shows a list of every VM associated with your app, detailing its state, region, and resource configuration.
09	`list_volumes`	Displays all persistent storage volumes for an app, including their size, status, and attached machine.
10	`restart_machine`	Forces a restart of an existing virtual machine to apply changes or recover from issues.
11	`start_machine`	Brings a stopped VM online, allowing it to boot and run its configured image again.
12	`stop_machine`	Shuts down a running machine's compute resources while preserving its configuration and data.
13	`update_machine`	Modifies the settings of an existing virtual machine, including changing the Docker image or resource allocations.

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 machines running in my web-api app.

Your web-api app has 3 machines: 'web-api-1' in iad (started, 1 CPU, 512MB), 'web-api-2' in fra (started, 1 CPU, 512MB) and 'web-api-3' in nrt (stopped). All running image registry.fly.io/web-api:v2.1.0.

U Start the stopped machine in Tokyo and restart the one in Frankfurt.

Done! I've started machine 'web-api-3' (nrt) — it's now booting. I've also restarted 'web-api-2' (fra) — it will stop and start again with its current configuration. Both machines should be healthy within a minute.

U Create a new 10GB volume called 'data' for my database app in Virginia.

Done! I've created a 10GB persistent volume called 'data' in the iad (Virginia) region for your database app. The volume is ready to be attached to a machine via its mount configuration.

Frequently Asked Questions

01 **How do I list all my deployed apps using the Fly.io Alternative MCP?**

You use `list_apps`. This command provides a master inventory, giving you the name and current status of every application your company has running on the edge network.

02 **Can I create a machine in a specific global region with Fly.io Alternative MCP?**

Yes, you can use `create_machine` to specify the desired region (like 'sjc' or 'fra'), ensuring your new compute resource meets low-latency requirements.

03 **What is the difference between starting and restarting a machine?**

Use `start_machine` if the VM was fully stopped. Use `restart_machine` when you want to apply configuration changes or recover from minor issues while keeping it active.

04 **Is there a way to check which volumes are attached to a machine?**

You first use `list_volumes` to see all available persistent storage. This list details the volume ID and, critically, the specific machine it is currently attached to.

05 **Does the Fly.io Alternative MCP help with updating resource size?**

Yes. If your app needs more memory or CPU, you use `update_machine` by providing a JSON body that specifies the new resource limits for the target VM.

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 `"flyio-alternative": { "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

Fly.io 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 Fly.io. 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	Fly.io MCP
Server ID	`019d843c-4ef1-71b5-b127-c49aba70afc8`
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/flyio-alternative.