MCP SERVER | NO CODE | CLOUD HOSTED

Fly.io Extended MCP

Control global infrastructure from conversation.

Fly.io Extended MCP connects your AI client directly to your cloud infrastructure. Provision apps, control machine lifecycles (start, stop, suspend), manage persistent volumes, and handle network certificates all from natural language commands.

A+ Quality Score 100/100

fly-io | cloud-computing | infrastructure-as-code | paas | serverless-machines

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

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

28 tools available
Cloud-hosted on Vinkius

Need to change something on your Fly.io setup? Instead of logging into a dashboard or running complex CLI scripts, you talk to your AI agent. This MCP gives your client full access to orchestrate global cloud resources through plain conversation. You can create new applications, scale up machines by changing their state, and even provision and expand storage volumes for your data. It handles everything from managing custom network certificates to requesting OpenID Connect tokens for secure identity. If you're building complex systems, this kind of integrated control is a game changer. Vinkius makes connecting all these services simple; just subscribe and get started working with your global deployment stack.

Core Capabilities

01 — Managing Applications

Create, list, or delete entire applications hosted on Fly.io.

02 — Controlling Machine States

Start, stop, suspend, update, and wait for specific machines to reach a desired operational state.

03 — Handling Storage Volumes

Create new volumes, list existing ones, or increase the size of current persistent storage.

04 — Securing Deployments

Request Let's Encrypt certificates or import custom PEM files for your services.

05 — Identity and Access

Get OpenID Connect tokens to verify secure workload identities across your infrastructure.

One Click on Vinkius — From Prompt to Execution

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

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

02 Your AI client authenticates with the token, gaining full access to the global API endpoints.

03 You issue a natural language command (e.g., 'Stop the web frontend machine'). The agent translates that request into the necessary cloud operations.

The bottom line is you write what you need done; this MCP handles the complex, multi-step process of talking to your cloud provider's APIs.

Built For

This tool is for DevOps Engineers and SRE teams who get frustrated by context switching—the constant jumping between terminals, dashboards, and Git. If you spend time copying IDs or running sequential scripts just to manage a simple outage, this MCP saves your sanity.

DevOps Engineer

Automating resource scaling, machine restarts, and volume management without ever leaving the chat interface.

SRE Team Member

Quickly inspecting infrastructure health, managing certificates during incident response, or updating machine configurations on the fly.

Full-stack Developer

Deploying new app instances and checking machine statuses directly from their code editor without leaving their IDE workflow.

What Changes When You Connect

01 Speed up incident response. Instead of manually checking multiple dashboards, you can ask your agent to retrieve the status and metadata for a specific machine using get_machine or list_machines. This means immediate visibility into critical systems.

02 Simplify resource provisioning. Need more space? Use extend_volume to increase storage size on a volume without logging into a separate console. Then, use create_volume if you're starting fresh.

03 Manage the full lifecycle in one place. You can delete an entire app with delete_app, or granularly remove just a machine using delete_machine and then clear up related volumes.

04 Secure your service identity fast. When rolling out new services, use create_acme_certificate to automatically get necessary SSL credentials, avoiding manual certificate renewal steps entirely.

05 Maintain operational continuity. You can suspend an expensive development environment with suspend_machine when no one's using it, and then easily bring it back online later with start_machine.

Real-World Applications

Debugging a production slowdown

An SRE notices high latency. Instead of SSHing into the server or clicking through monitoring tools, they ask their agent to run get_machine and list_volumes for that app. The response immediately provides detailed metadata on machine health and volume attachment points.

Rolling back a bad deployment

A full-stack developer pushes faulty code. They realize they need an older version of the data. They use list_volume_snapshots to see available backups, then initiate a restore process by calling create_volume with the snapshot ID.

Scaling for peak traffic

The DevOps team anticipates a massive user influx. Instead of manually increasing resource limits, they instruct their agent to use update_machine on multiple machines and then call start_machine across all nodes to ensure capacity is ready.

Decommissioning an old service

The team retires a legacy microservice. They don't want any remnants left behind, so they use delete_app first to wipe the app, followed by list_volumes and delete_volume for all associated storage.

Patterns to Avoid

Trying to guess IDs

✗ AVOID

The user knows a machine exists but doesn't remember its exact ID. They waste time searching through old logs or multiple tabs trying to find the right unique identifier.

✓ INSTEAD

First, use list_machines to see all available instances in your app. Then, you can reference the correct name or status when you run commands like stop_machine or get_machine.

Over-managing state

✗ AVOID

The user tries to start a machine that's already running and fails because they didn't check the current state. They end up manually trying multiple redundant API calls.

✓ INSTEAD

Always use get_machine first. This tool gives you the definitive, real-time status, letting your agent know if stop_machine or suspend_machine is even necessary.

Confusing resources

✗ AVOID

The user wants to increase storage but runs a command that only affects the application's network settings, leaving data unexpanded.

✓ INSTEAD

To expand storage capacity, you must use extend_volume. This tool specifically targets increasing the size of existing persistent volumes.

The Right Fit

Use this MCP if your daily work involves managing cloud infrastructure resources that require multiple steps or specific state transitions, like starting a machine, then updating its configuration, and finally verifying a certificate. You need control over the full lifecycle—from app creation (create_app) to deletion (delete_app). Don't use it if you simply need to read public data about Fly.io; for that, consult official documentation. If your goal is just basic code

execution without infrastructure interaction, your AI client's native shell commands are fine. But if the task involves provisioning, scaling, or managing persistent state (volumes/certificates), this MCP is essential because it bundles all those complex actions into simple, natural language calls.

Managing Cloud Infrastructure Today Is a Chore

Today, deploying or updating anything on Fly.io means context switching: checking the dashboard for machine status, running CLI commands to update configurations, manually requesting certificates in another tab, and then finally expanding storage via a separate console. It's a painful sequence of logins, copy-pasting IDs, and waiting for confirmation screens.

With this MCP, you tell your agent what you need done—for example, 'Scale up the database machine and renew its certificate.' The system handles the entire workflow: it checks the current state, calls update_machine, then runs create_acme_certificate. You just get a clean confirmation.

Fly.io Extended MCP Gives Full Control Over Your Stack

The manual steps that vanish include manually running `fly apps list`, then needing to find the machine ID to run `fly machines status`, and subsequently having to jump over to a different portal just to renew an SSL certificate.

Now, you can ask your agent to orchestrate all three actions in one prompt. You manage everything—app listing (list_apps), machine state transitions (start_machine/stop_machine), and security credentials (get_certificate)—without leaving the conversation.

Fly.io Extended MCP: 25 Tools

Use these tools to programmatically control every aspect of your cloud infrastructure, from deploying new apps to managing machine lifecycles and storage volumes.

#	TOOL	DESCRIPTION
01	check_certificate	Triggers a DNS validation check to confirm a certificate's status.
02	create_acme_certificate	Requests a new, valid Let's Encrypt SSL/TLS certificate for an application.
03	create_app	Creates and initializes a brand-new Fly.io application within your account.
04	create_custom_certificate	Imports a certificate using its own PEM file, useful for internal services.
05	create_lease	Creates temporary access leases for specific machines to manage resource usage.
06	create_machine	Builds and deploys a new Fly machine instance using a specified configuration image.
07	create_volume	Allocates and initializes a brand-new, persistent storage volume for your app.
08	delete_app	Permanently removes an entire Fly application from your account.
09	delete_certificate	Removes both the hostname and the associated digital certificate records.
10	delete_machine	Deletes a specific machine instance, taking it out of service entirely.
11	delete_volume	Permanently deletes an allocated storage volume and its data.
12	extend_volume	Increases the total available size of an existing persistent storage volume.
13	get_app	Retrieves all current operational details for a specific Fly application.
14	get_certificate	Fetches detailed information about an existing certificate.
15	get_machine	Retrieves the current status and configuration details for a specific machine.

#	TOOL	DESCRIPTION
16	`get_volume`	Provides detailed metrics and metadata about an existing storage volume.
17	`list_apps`	Lists all applications currently deployed under your account namespace.
18	`list_certificates`	Retrieves a list of every certificate tied to a specific application.
19	`list_machines`	Lists all running and suspended machine instances within an app.
20	`list_volume_snapshots`	Shows you available snapshots of your persistent volumes for backup/rollback purposes.
21	`list_volumes`	Retrieves a list of all storage volumes attached to an application.
22	`release_lease`	Releases temporary access leases, freeing up allocated machine resources.
23	`request_oidc_token`	Generates a secure OpenID Connect token needed for workload authentication.
24	`start_machine`	Initiates the startup sequence, bringing a suspended or stopped machine back online.
25	`stop_machine`	Gracefully stops a running machine, preserving its current state for later restart.
26	`suspend_machine`	Temporarily suspends a machine to save compute resources without losing its configuration.
27	`update_machine`	Modifies the core configuration parameters of an existing machine instance.
28	`wait_machine`	Pauses execution until a specified machine reaches a desired operational state (e.g., running).

See It in Action

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

U List all my Fly.io apps in the 'personal' organization.

I've retrieved your apps for the 'personal' organization. You have 3 active apps: 'my-api-server', 'web-frontend-prod', and 'db-primary'. Would you like to see the machines running in any of these?

U Start the machine with ID 9185210b6e4d8p in the app 'my-api-server'.

Starting machine 918... The request was successful. The machine is now transitioning to the 'started' state.

U Create a 10GB volume named 'data_vol' in the 'ams' region for app 'db-primary'.

I've initiated the creation of a 10GB volume 'data_vol' in Amsterdam (ams) for 'db-primary'. The volume ID is vol_z9rj5k2m1l. It is now ready to be attached to a machine.

Frequently Asked Questions

01 **How do I get a new SSL certificate using Fly.io Extended MCP?**

You use create_acme_certificate to request a fresh Let's Encrypt certificate for your application. The agent handles the validation process and attaches it automatically.

02 **What is the difference between stop_machine and suspend_machine with Fly.io Extended MCP?**

Stop_machine gracefully shuts down the instance, preserving its state. Suspend_machine saves compute resources entirely, which is useful if you're just pausing development work.

03 **Can I increase storage size using the Fly.io Extended MCP?**

Yes, use extend_volume to increase the allocated space of an existing volume without having to delete and recreate the entire data set.

04 **Which tool should I use if I want to see all my deployed applications?**

Use list_apps. This single command retrieves a comprehensive overview of every application currently running under your account namespace.

05 **Does Fly.io Extended MCP handle deleting the entire app and its data?**

Yes, delete_app removes the entire application container. Remember that you may still need to manually manage associated volumes using delete_volume for complete cleanup.

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-extended": { "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	`019e3899-27c5-700d-9992-5117e2138543`
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-extended.