MCP SERVER | NO CODE | CLOUD HOSTED

Mux MCP

Control your entire video content pipeline conversationally.

Mux MCP connects your video infrastructure directly to your AI agent. Instantly manage everything from creating new assets and handling secure uploads to monitoring live streams and pulling view analytics. Control complex video workflows using natural conversation, without touching a dashboard.

A+ Quality Score 100/100

video-infrastructure | live-streaming | asset-management | video-encoding | playback-api | media-processing

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

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.

Mux MCP

11 tools available
Cloud-hosted on Vinkius

You'll take full control of all your video content and streaming operations right through your chat client. Forget jumping between dashboards just to check an asset status or generate metadata. This MCP lets you talk to your video infrastructure like it’s a simple database query.

Need to launch a new stream? Just ask for it. Need to see how many views a piece of content got last week? Your agent pulls that data instantly. You can even trigger the creation of an asset just by giving it a public URL, and your agent handles the whole process. When you connect this MCP through Vinkius, you get access to a comprehensive catalog of tools, making sure video management is always available wherever you work.

It covers everything: listing all existing files, checking which are finished processing, setting up direct upload sessions for large files, and monitoring live stream health in real-time. It makes your entire media pipeline conversational.

Core Capabilities

01 — Manage Asset Lifecycle

Create, list, retrieve details for, or delete any video asset using simple commands.

02 — Handle Live Streaming

Set up new live streams and check the current configuration or status of existing broadcasts.

03 — Secure Video Uploading

Generate secure, temporary endpoints for users to upload large video files directly to your system.

04 — Monitor Performance and Data

Fetch recent view metrics or pull complete metadata records for both assets and live streams.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your unique Mux Token ID and Secret.

02 Authorize your AI client, giving it permission to interact with video resources.

03 Ask your agent to perform a task, such as 'List all my assets that are ready for playback.'

The bottom line is you control complex media operations by talking to the system instead of clicking through multiple web tabs.

Built For

Video developers, content platform managers, and broadcast engineers who get frustrated having to switch between Mux's dashboard, their chat app, and a terminal just to run simple checks. You need automated workflows that speak your language.

Video Developer

Needs to quickly check if an asset has finished encoding without logging into the Mux platform.

Content Platform Manager

Uses the tool to automate the creation of new video assets or set up direct upload endpoints when deploying content across multiple channels.

Broadcast Engineer

Monitors live stream configurations and health status in real-time, getting immediate alerts through their communication tools instead of watching a dedicated monitoring dashboard.

What Changes When You Connect

01 Eliminate dashboard hopping. Instead of manually checking the Mux UI to see if a file is ready, just ask your agent to list assets and check their status; it's immediate.

02 Speed up deployment by automating asset creation. You can initiate new video assets simply by giving your agent a source URL and asking it to create the resource.

03 Handle huge files securely. Use the direct upload tool to generate temporary, secure links, letting users bypass traditional file sharing methods when uploading large videos.

04 Keep track of performance without logging in. Get recent views metrics instantly, allowing you to know exactly how well your content is performing right inside your chat window.

05 Simplify live broadcasting setup. Your agent can list available streams or create a brand new one just by talking to it, handling all the underlying configurations.

Real-World Applications

Checking asset readiness after an update

A video developer finishes uploading raw footage and asks their agent to list assets. The agent then calls get_asset for the new file, confirming it's still 'processing.' They ask the agent to check its status again in 10 minutes, saving them from having to manually refresh a dashboard.

Generating campaign content quickly

A marketing team member needs three video assets for an ad rollout. They tell their agent which source URLs are available. The agent uses create_asset multiple times, then lists_assets to give the manager a confirmation list of all new IDs.

Diagnosing stream failure

A broadcast engineer notices a live feed dropping connections. They ask their agent to get_live_stream details for the main channel ID, immediately pinpointing if the issue is with the configuration or credentials, and not just general network failure.

Analyzing content reach post-launch

A content manager posts a video and later asks their agent to get_recent_views. The agent pulls the analytics data right away, allowing them to decide if they need to promote it or rethink its topic based on real numbers.

Patterns to Avoid

Manual API scripting

✗ AVOID

Writing a complex Python script that uses the Mux SDK, managing tokens, error handling, and sequential calls just to list assets or check views.

✓ INSTEAD

Connect this MCP. Instead of writing boilerplate code, you simply ask your agent: 'List all video assets and show me their status.' The tool handles the API interaction for you.

Dashboard Overload

✗ AVOID

Spending minutes clicking through Mux's web interface—navigating to 'Analytics,' then finding the 'Views' tab, and finally copying the numbers.

✓ INSTEAD

Use your agent. Just prompt it with: 'What were my recent video views?' The MCP immediately retrieves that specific metric without you having to click anything.

Ignoring asset dependencies

✗ AVOID

Trying to delete an asset or modify a live stream configuration without first fetching its complete metadata, leading to API errors.

✓ INSTEAD

Always use the get_asset tool first. This fetches all required metadata for a given ID before you attempt destructive actions like using the delete_asset tool.

The Right Fit

Use this MCP if your primary bottleneck is switching between tools or dashboards to manage video workflows. You need natural language control over core functions like creating assets, monitoring streams, and pulling analytics. It's perfect for devs who want to prototype a media pipeline without writing boilerplate API code.

Don't use this if you only need to view a static list of IDs (you can do that with basic search). Also, don't use it if your primary need is integrating Mux data into a totally different system like a CRM or ERP; for those cases, look for an integration focused on structured record mapping. This MCP excels at controlling the *Mux platform itself* via conversation.

Dealing with Video Assets Feels Like Running Around in Circles

Today, managing video assets means logging into Mux's dashboard. You go to 'Assets,' then you scroll through maybe fifty files just to check if the one you care about finished processing. If you need analytics, you have to click another tab, copy down a view count, and paste it somewhere else. It’s constant clicking and context switching.

With this MCP connected to your agent, that whole process vanishes. You simply tell your AI client: 'Check the status of asset-123.' The answer comes back instantly in your chat window. Your workflow stays contained, fast, and conversational.

Get Full Control with Mux MCP

You no longer need to manually initiate a live stream or list uploads. You can ask your agent to create a new live stream configuration immediately, or use the list_direct_uploads tool to see exactly which secure endpoints are active and when they expire.

The difference is control. This MCP gives you conversational access to every core function of Mux—from creating assets to getting recent views—all without ever leaving your preferred client.

Mux MCP: 12 Tools for Media Management

These tools allow your agent to execute every core video operation—from asset creation to viewing analytics—using simple conversation commands.

#	TOOL	DESCRIPTION
01	`create_asset`	Create a new video asset
02	`create_direct_upload`	Create a direct upload session
03	`create_live_stream`	Create a new live stream
04	`delete_asset`	Delete a video asset
05	`delete_live_stream`	Delete a live stream
06	`get_asset`	Get specific asset details
07	`get_live_stream`	Get specific live stream details
08	`get_recent_views`	Get recent video views analytics
09	`list_assets`	List Mux video assets
10	`list_direct_uploads`	List direct video uploads
11	`list_live_streams`	List Mux live streams

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 video assets and show their status.

I've retrieved your video assets. You have 12 assets, including 'asset-123' (Status: ready) and 'asset-456' (Status: processing). Shall I check if 'asset-456' is finished now?

U Create a new video asset from https://example.com/video.mp4.

I've initiated the creation of a new video asset from the URL provided. The new Asset ID is 'new-asset-789' and its current status is 'preparing'. I'll let you know when it's ready for playback.

U Check recent views for my videos.

Fetching recent analytics... You had 156 views in the last 15 minutes, with a 98% player success rate. Shall I breakdown the views by country?

Frequently Asked Questions

01 **How do I use the Mux MCP to check an asset's status?**

Use the list_assets tool and then specify the ID of the video you want checked. Your agent will retrieve the current details, letting you know if it's still processing or ready for playback.

02 **Can the Mux MCP handle live stream setup?**

Yes, you can use create_live_stream to set up a new broadcast and list_live_streams to see all your existing configurations. It keeps your streaming monitoring centralized.

03 **What is the best way to upload large videos using Mux MCP?**

Always use the create_direct_upload tool first. This generates a temporary, secure URL that allows massive files to be uploaded directly without burdening your internal infrastructure.

04 **Does the Mux MCP give me performance data?**

Absolutely. The get_recent_views tool fetches analytics showing view counts and other metrics, so you can track content reach instantly through natural language prompts.

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 `"mux": { "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

Mux 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 Mux. 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	Mux MCP
Server ID	`019d75d9-7ca7-7101-a6bd-f4e337a258a7`
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/mux.