

MCP SERVER

NO CODE

CLOUD HOSTED

Uploadcare MCP

Manage media assets and file infrastructure via chat.

Uploadcare lets you manage your entire media pipeline directly through natural language conversation. Connect it to your AI agent to list files, check technical metadata, organize collections, and execute bulk operations like deleting or storing assets without ever touching a dashboard.

A+ Quality Score 98.33/100

file-upload

cdn

media-processing

asset-management

cloud-infrastructure

image-optimization



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

Uploadcare MCP

10 tools available

Cloud-hosted on Vinkius

This MCP connects your Uploadcare account, giving your AI agent full control over your file infrastructure. Instead of logging into the web dashboard for every task—checking dimensions, verifying CDN URLs, or monitoring storage quotas—you simply ask your agent to handle it.

It lets you manage assets from creation to archival. You can list all existing files and inspect their technical metadata on demand. If you need to clean house, you don't have to manually select hundreds of outdated images; you just tell the agent to batch delete them. For large-scale data projects, it also handles copying files to external storage like S3 buckets, or marking temporary assets as permanently stored when they're ready for prime time.

If your workflow involves any kind of media distribution—from product photography to massive content libraries—this MCP is the control center you need. It's available here on Vinkius so you can connect it easily from Claude, Cursor, or any other AI-compatible client.

Core Capabilities

01 — Inspect File Metadata

Get technical details for a specific file, including its size, dimensions, and direct public CDN URL.

03 — Perform Bulk Operations

Mark multiple temporary files as permanent storage using batch operations, or irreversibly delete large sets of outdated content at once.

05 — Transfer Files

Copy existing media files manually to external destinations like S3 buckets or local storage targets.

02 — Manage Collections

List entire groups of files or collections, and inspect exactly which individual assets belong to them.

04 — Monitor Project Usage

Retrieve current project-level data on your account's total stored capacity and bandwidth consumption limits.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP and provide your Uploadcare Public and Secret Keys.
- 02 Connect the credentials to your preferred AI client (Claude, Cursor, etc.).
- 03 Use natural language prompts to instruct your agent to perform file actions or retrieve data.

The bottom line is you treat your entire media infrastructure like a conversation; you don't interact with confusing dashboards.

Built For

Anyone managing large volumes of digital assets—from small marketing teams to enterprise DevOps groups. It's for the engineer who can't afford to spend hours clicking through a dashboard just to check if an asset is ready, or the content manager drowning in outdated images needing mass cleanup.

DevOps Engineer

Monitoring project storage limits and bandwidth usage, or automating the copying of files from Uploadcare into external cloud buckets like S3.

Software Engineer

Debugging asset upload issues by quickly retrieving specific file technical metadata (like dimensions) or finding a direct CDN URL without opening the console.

Content Manager

Cleaning up unused assets and permanently deleting massive batches of temporary files using simple natural language commands.

What Changes When You Connect

- 01 You stop clicking through dashboards. Your agent handles complex tasks, such as running `get_file_details` to check a CDN URL or dimensions instantly.

-
- 02 Mass cleanup is effortless. Instead of manually deleting files, you use batch operations like `batch_delete_files` on outdated assets in seconds.

 - 03 Always know your limits. The agent calls `get_project_info` so you can monitor storage and bandwidth usage without guessing if you're running low.

 - 04 Automate transfers. Use `copy_file` to automatically move necessary media from Uploadcare into external systems like S3 buckets for backup or processing.

 - 05 Organize assets logically. You can list all file groups using `list_file_groups` and then check the contents of any group with `get_group_details`.
-

Real-World Applications

Auditing old content libraries

The Content Manager needs to figure out how much space is taken up by temporary assets. They ask the agent, and it calls `get_project_info` to report current storage limits. Then they identify 50 unused images via `list_files` and use `batch_delete_files` to clear the quota.

Archiving project resources

The DevOps Team identifies a set of temporary images that should be kept long-term. They instruct the agent using `batch_store_files`, moving those files from volatile temporary status to permanent, reliable storage.

Preparing a file for external ingestion

The Software Engineer needs an asset ready in another system. They ask the agent to check the file's technical metadata using `get_file_details`, confirm its CDN URL, and then use `copy_file` to send it directly to their staging S3 bucket.

Inspecting a known asset group

The Product Team needs to confirm which files belong to 'Q3 Campaign Images'. They ask the agent to call `list_file_groups` to find the group name, then use `get_group_details` to verify all included assets are present.

Patterns to Avoid

Trying to copy files manually

✗ AVOID

Logging into the Uploadcare dashboard and clicking through multiple file listings, then finding the download link for each asset one by one.

✓ INSTEAD

Tell your agent directly: 'Copy these specific assets to my S3 bucket.' The tool uses ``copy_file`` to execute the transfer in a single conversational step.

Forgetting to check file status

✗ AVOID

Assuming an image is permanently saved when it's actually just temporary, leading to data loss or difficulty retrieving metadata.

✓ INSTEAD

Always ask your agent to run ``get_file_details`` first. This confirms the technical metadata and current storage state of the asset before any operation.

Deleting files without knowing group membership

✗ AVOID

Wiping out assets because they look old, but accidentally deleting a file that was part of a critical 'Group' collection needed for another project.

✓ INSTEAD

Before mass deletion, use ``list_file_groups`` to see existing collections and then run ``get_group_details`` to confirm what you are about to lose.

The Right Fit

Use this MCP if your workflow revolves around the lifecycle of digital media assets—uploading them, tracking their metadata, organizing them into logical groups, or cleaning up massive amounts of temporary junk. It's essential when you need programmatic control over CDN storage without opening a browser. Don't use it if you just need basic file *uploads* (that's for standard SDK connections). If your problem is purely about optimizing AWS bucket policies or managing permissions outside the media content itself, then look at dedicated cloud governance tools instead.

The hassle of manual asset management

Today, handling a large library of images means navigating complex dashboards. You have to click into groups, manually check technical metadata, and then copy-paste the CDN URL if you need it elsewhere. If you find outdated assets, deleting them involves finding every single file ID and running multiple commands just to clear out junk.

With this MCP, those clicks disappear. You simply tell your agent what needs doing—like listing all files or checking project storage limits—and get the data instantly. Your AI client does the heavy lifting behind the scenes.

Uploadcare gives you full control over assets

You no longer have to manually check if a file is permanently stored or what its dimensions are. You can get all that information, including the raw public CDN URL, using `get_file_details`—all from a single chat prompt.

The difference now is control. Your agent makes your entire media infrastructure accessible through conversation, letting you manage assets as if they were just another API call.

Uploadcare: 10 File Management Tools

These tools allow you to perform every operation needed for media asset management, from checking file details to executing massive bulk deletions.

#	TOOL	DESCRIPTION
01	<code>batch_delete_files</code>	Permanently removes multiple selected files from the project in one action. This change cannot be undone.
02	<code>batch_store_files</code>	Designates a group of temporary files, marking them as permanently stored and active assets.
03	<code>copy_file</code>	Copies an existing file to specified local or remote storage targets like S3 buckets.
04	<code>delete_file</code>	Permanently removes a single file and all its derived variants from Uploadcare. This action is irreversible.
05	<code>get_file_details</code>	Retrieves specific technical information, like dimensions or mime type, for one asset.
06	<code>get_group_details</code>	Provides detailed information about a specific file group, including its contents and status.
07	<code>get_project_info</code>	Pulls project-level metrics, showing current storage usage and bandwidth consumption totals.
08	<code>list_files</code>	Retrieves a paginated list of all files currently stored in the Uploadcare project.
09	<code>list_file_groups</code>	Lists all immutable file collections (groups) that have been created within the project.
10	<code>store_file</code>	Marks a single temporary file, moving it to permanent storage status.

See It in Action

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

U What is our current project storage and bandwidth usage?



Based on your project info, you currently have 15.2 GB stored across 45,120 active files. Your bandwidth for this billing period is at 230 GB. This places you at 85% of your current plan limits. Would you like me to inspect and list the largest files so we can batch delete some test data?

U Can you check the dimensions and CDN URL for file UUID `9cd83...`?



I checked the asset. It's a `hero-banner.jpg` (mime_type: image/jpeg) weighing 2.4MB. It is currently marked as permanently stored. Its dimensions are 3840×2160 pixels. The raw public CDN access url is `https://ucarecdn.com/9cd83.../`. Let me know if you want me to copy this to an S3 bucket.

U Batch delete these 4 outdated temporary images: `e33b...`, `f55a...`, `8c11...`, `ab99...`.



I've successfully executed the `batch_delete_files` command. All 4 assets (`e33b...` , `f55a...` , `8c11...` , `ab99...`) and their related CDN variants have been irrevocably deleted from your Uploadcare project. Your storage limits have been safely reduced.

Frequently Asked Questions

01 Can the Uploadcare MCP help me find file metadata?

Yes, it can. You use `'get_file_details'` to retrieve technical metadata for any specific asset, including its dimensions and CDN URL.

02 How do I delete a lot of old images with Uploadcare MCP?

You run the `batch_delete_files` tool. This permanently removes multiple assets in one operation, which is much faster than deleting them individually.

03 Does this MCP monitor my storage quota?

Yes. The agent calls `get_project_info` to provide real-time metrics on your total account storage and bandwidth usage limits.

04 What is the difference between listing files and listing groups with Uploadcare MCP?

Using `list_files` shows every individual asset in your project. However, you use `list_file_groups` to see immutable collections, which organize those assets logically.

05 Can I move files out of Uploadcare using the MCP?







Yes. The agent uses the `copy_file` tool to copy existing media directly to external storage targets like S3 buckets.

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 <code>"uploadcare": { "url": "..."} </code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Uploadcare 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](https://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 Uploadcare. 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	Uploadcare MCP
Server ID	019d7619-e963-72b8-a823-0c17b05889e4
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/uploadcare.