MCP SERVER | NO CODE | CLOUD HOSTED

ImageKit MCP

Manage Media Assets and Purge Caches with Conversational AI

ImageKit (Media Optimization & DAM) connects your AI client directly to ImageKit, giving you full control over cloud-native media management and real-time image optimization. You can list all visual assets, audit EXIF data like focal length and ISO, purge CDN caches with precise URLs, or update custom metadata fields in bulk.

A+ Quality Score 100/100

media-optimization | cdn | image-processing | asset-management | real-time-transformation

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/imagekit-media-optimization-dam

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.

ImageKit (Media Optimization & DAM) MCP

10 tools available
Cloud-hosted on Vinkius

When you connect your AI agent to this MCP, it gives you a natural conversation layer over complex media infrastructure. Instead of jumping through multiple developer consoles to manage image assets, you simply ask the question. You can list every file uploaded to your vault and retrieve deep details, including EXIF data or custom tags. Need fresh images? Tell your agent to purge specific URLs across the global CDN network. It handles that request instantly and monitors the status for you. Furthermore, you gain control over metadata, allowing you to patch asset tags or create new fields to map business-specific variables directly onto your assets. This whole process is managed through a single connection point on Vinkius, letting your agent handle everything from auditing file properties to irreversibly deleting old files.

Core Capabilities

01 — List and audit visual assets

The MCP lists all uploaded media files and retrieves structural data like EXIF metadata or AI-generated tags.

02 — Purge content delivery networks (CDN)

You can trigger cache invalidation for specific URLs across the global edge network to ensure users see the latest version of an image.

03 — Update and structure metadata

The MCP allows you to update tags or create custom data fields on assets in bulk, keeping your digital asset management system organized.

04 — Retrieve file properties

You can pull specific structural details from an image, like its ISO rating or focal length.

05 — Remove files permanently

The MCP executes irreversible deletion of assets to optimize cloud storage and clean up old media libraries.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/imagekit-media-optimization-dam** — connect your AI agent in three steps.

01 Subscribe to the ImageKit MCP using your private key.

02 Your AI client accesses all available tools, allowing you to run commands like listing assets or checking metadata status.

03 The tool executes the action against your live ImageKit account and reports the outcome back to your agent.

The bottom line is that you get a single conversational interface for managing complex media infrastructure.

Built For

Anyone who spends time in developer consoles dealing with image pipelines, front-end performance issues, or large media libraries. This MCP saves the content ops engineer from manually auditing metadata and stops the frontend developer from running multiple scripts just to clear a CDN cache.

Frontend Developer

Verifying image delivery URLs and purging CDN caches for assets right inside their IDE without switching context or running CLI commands.

Content Operations Specialist

Auditing metadata across thousands of images, grouping them by structural properties (like ISO or focal length), and ensuring consistency in the media library.

DevOps Engineer

Integrating asset lifecycle management into CI/CD pipelines by programmatically checking file existence and initiating cache purges after deployment.

What Changes When You Connect

01 Save time auditing metadata. Instead of manually checking hundreds of files in a GUI, you can use the `get_exif_metadata` tool to automatically retrieve technical properties like ISO or focal length for compliance checks.

02 Ensure fresh content delivery every time. When assets change, run `purge_cdn_cache` via your agent to invalidate URLs instantly across all global nodes, eliminating stale image problems.

03 Maintain a clean asset database using bulk operations. Use `wipe_batch_assets` or `wipe_media_asset` for controlled, irreversible cleanup of old or unused media files.

04 Extend data structure easily. If your business needs to track custom variables (like 'Client ID' or 'Campaign Tag'), use the MCP to create and manage these fields via `create_custom_schema`.

05 Get a complete picture of your assets instantly. The `list_media_files` tool gives you an immediate inventory, while `get_file_details` provides the deep dive into any specific asset ID.

Real-World Applications

Fixing stale hero images after a site redesign

The marketing team updated the main banner image but noticed some users saw the old version. They ask their agent to `purge_cdn_cache` for the specific URL, and the agent confirms the invalidation status using `get_purge_status`. The problem is solved in seconds.

Auditing product photos before a collection launch

A content specialist needs to verify that all 50 new product shots were taken with a consistent aperture (f/2.8). They ask their agent to run `get_exif_metadata` on the entire batch, quickly flagging any images that don't match the required structural properties.

Cleaning up old draft content

A developer is optimizing storage and finds thousands of abandoned draft assets. Instead of manually deleting them, they use `wipe_batch_assets` to safely remove all files tagged as 'DRAFT' from the media vault.

Mapping product data to images

The e-commerce team needs to ensure every image is associated with a specific internal product SKU. They use `create_custom_schema` to add an 'SKU' field and then use `patch_file_details` in bulk to tag all existing assets.

Patterns to Avoid

Treating the MCP like a simple file explorer.

✗ AVOID

Trying to just 'look at' files without specifying what metadata you need. This often leads to vague, incomplete lists that don't tell you if the image is ready for production.

✓ INSTEAD

Always ask for specific data points. Use `get_exif_metadata` to get technical camera specs, or use `list_custom_fields` first, then reference those fields when using `patch_file_details`. Be precise about what you're checking.

Assuming the CDN cache automatically updates.

✗ AVOID

A developer pushes new content and just assumes it's live. They check the image in their browser, but sometimes they still see the old version because of a cached edge node.

✓ INSTEAD

Never assume. After uploading or updating an asset, immediately tell your agent to run `purge_cdn_cache` for that URL. Then, use `get_purge_status` later to confirm the purge worked.

Manually tracking file status across multiple tools.

✗ AVOID

Juggling a spreadsheet of asset IDs and having to run separate checks in different consoles just to know if the assets are valid or what their tags are.

✓ INSTEAD

Let your agent handle the orchestration. Use `list_media_files` for inventory, then use `get_file_details` to gather all necessary context into one single output.

The Right Fit

Use this MCP if your core problem involves managing media assets at scale and maintaining consistency across multiple systems. Specifically, if you need to read technical properties (EXIF), manage custom business data (metadata patching/schemas), or guarantee content freshness (CDN purging). Don't use it if your only goal is

simple file storage; there are dedicated object storage solutions for that. If all you need to do is list files without any metadata audit, a basic listing tool works, but this MCP gives you the critical control layer needed by developers and operations teams.

This tool excels when you need to perform an action (like purging) AND verify its status (`get_purge_status`) using the asset's unique ID. If your process is purely about writing new images without ever touching metadata or caches, then this MCP offers too much complexity, but for most professional workflows, it’s essential.

The headache of stale media and disorganized assets

Today, updating a single image across a large website is a multi-step nightmare. You upload the new asset, then you have to remember to go into the CDN dashboard and manually paste every relevant URL to purge the cache. If you forget even one, users see old content. On top of that, finding out if an asset has the right technical specs—like being shot at 300mm focal length—means jumping between image viewers and metadata sheets, copying data point by painful data point.

With this MCP, you simply tell your agent what needs to happen. You ask it to purge a batch of URLs, and it handles the entire invalidation process. Need to check specs? Ask for `get_exif_metadata` . The whole manual, copy-pasting workflow collapses into one single conversation.

Get full control with ImageKit's structured metadata

Previously, if you needed to track a business variable—say, 'Q3 Campaign'—you were stuck. You might use external spreadsheets or rely on unstructured filenames that quickly become messy and impossible to query reliably at scale.

Now, you can use the MCP to define these variables using `create_custom_schema` . This means your assets are no longer just files; they're structured data points. Your agent sees them as organized fields, making searches reliable and reports accurate.

ImageKit (Media Optimization & DAM) with 10 Tools

Use these tools to perform complex actions like listing files, checking technical metadata, and purging CDN caches directly through your AI client.

#	TOOL	DESCRIPTION
01	`create_custom_schema`	Creates new, structured metadata fields on your assets for tracking specific project data points.
02	`wipe_batch_assets`	Deletes multiple specified media files in a batch process to clean up storage space.
03	`wipe_media_asset`	Irreversibly deletes a single, specific asset from the system.
04	`get_file_details`	Retrieves detailed information about any given file ID, including its full path and status.
05	`get_exif_metadata`	Extracts technical camera data from an image, such as the dimensions or focal length used when it was shot.
06	`get_purge_status`	Checks if a CDN cache purge request has successfully completed across the network.
07	`list_custom_fields`	Displays all the custom metadata fields currently defined for your assets, helping you see what data is tracked.
08	`list_media_files`	Lists every visual asset available in your ImageKit vault to give you a complete inventory count.
09	`purge_cdn_cache`	Forcibly removes cached versions of assets at specified URLs, making sure users see the most recent content immediately.
10	`patch_file_details`	Modifies existing metadata tags or attributes for a file without changing the image itself.

See It in Action

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

U List the last 10 images uploaded to my ImageKit vault

I've retrieved the 10 most recent assets. Highlights include 'banner_hero.jpg', 'product_01_alt.webp', and 'logo_transparent.png'. Would you like to see the detailed metadata or tags for any of these?

U Purge the cache for this URL: https://ik.imagekit.io/myproject/header.jpg

Cache purge request sent successfully. Request ID: 'pur_98765'. The Edge CDN nodes will be invalidated shortly. I can check the purge status for you in a few minutes.

U Show me the focal length and dimensions for file ID 'file_12345'

EXIF metadata for file_12345: Dimensions: 4032×3024, Focal Length: 26mm, ISO: 100, Aperture: f/1.8. It was captured on an iPhone 13 Pro. Would you like to add an 'Apple' tag to this asset?

Frequently Asked Questions

01 **How do I check if a CDN cache purge actually worked with ImageKit MCP?**

You use the `get_purge_status` tool. This function queries the global network to confirm whether your previous invalidation request has completed successfully across all edge nodes.

02 **Can I bulk update tags using the ImageKit MCP?**

Yes, you can use the `patch_file_details` tool. This lets you modify asset tags or metadata fields for multiple files at once without having to edit every single one individually.

03 **What is the difference between wiping a batch and wiping one file with ImageKit MCP?**

Use `wipe_media_asset` when you need to remove just one specific image. If you are deleting many files, always use `wipe_batch_assets` for an automated check that routes explicit disk removals.

04 **Does ImageKit MCP help with SEO?**

Indirectly. By allowing you to update custom metadata fields and tags using the MCP, you can ensure assets contain necessary structured data points needed for internal search or external indexing.

05 **Can I get technical camera specs with ImageKit MCP?**

Yes, run `get_exif_metadata` to retrieve structural image properties like the focal length, ISO rating, and dimensions right from the file's data.

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 `"imagekit-media-optimization-dam": { "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

ImageKit (Media Optimization & DAM) 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 ImageKit (Media Optimization & DAM). 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	ImageKit (Media Optimization & DAM) MCP
Server ID	`019d75b8-55cc-71e9-ae9d-9a517044523f`
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/imagekit-media-optimization-dam.