MCP SERVER | NO CODE | CLOUD HOSTED

Immich MCP

Manage your entire media archive conversationally.

Immich MCP connects your self-hosted media library to any AI agent, giving you full control over photos and videos through natural conversation. You can list assets, create albums, manage users, or check the server status without opening a single tab.

A+ Quality Score 100/100

self-hosted | photo-management | media-library | asset-management | backup | digital-assets

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/immich-google-photos-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.

Immich (Google Photos Alternative) MCP

26 tools available
Cloud-hosted on Vinkius

Take complete command of your private photo and video archive using this MCP. It lets your AI client interact directly with every part of your self-hosted Immich instance—everything from individual photos to complex user structures. Instead of digging through dashboards, you talk to the library itself. Need a specific image? Just ask for it. Want to group pictures taken last summer? Tell your agent to create an album and populate it automatically. You can organize assets by listing every person recognized in the images or creating new tags across entire collections. When you connect this MCP via Vinkius, you bring that massive catalog of media management tools directly into your chat window, letting you manage all your data without leaving your workspace.

Core Capabilities

01 — Manage Assets and Media

You can list all photos or videos in the library, fetch specific details about an asset, update existing media records, delete old files, or even upload new media directly.

02 — Organize Albums and Collections

Create brand new albums, view detailed information for any existing album, update its contents, or remove it entirely when the memories are archived.

03 — Handle People and Tags

List all recognized people in your photos, get details on a specific person's profile, create new identity records, delete old ones, or manage custom tags across the entire library.

04 — Manage Users and Accounts

View who has access to the system by listing all users, creating new accounts, updating user profiles, or removing user access completely.

05 — Monitor System Health

Get instant information on your Immich server's current version and operating status without logging into the admin panel.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your specific Immich Instance URL along with a valid API Key.

02 Your AI agent uses those credentials to connect directly to your private media library's backend.

03 You simply type natural language requests (e.g., 'Create an album for the 2024 trip') and the MCP executes the required actions.

The bottom line is that you treat your entire complex, self-hosted photo archive like a simple conversation with an expert assistant.

Built For

This is for anyone who manages large volumes of private media and hates clicking. If you're constantly jumping between the web UI, spreadsheets, and administrative dashboards just to categorize photos or check user access, this MCP saves your sanity.

Digital Asset Manager

Uses the MCP to systematically audit photo libraries. They can run list_assets to see everything, then use get_asset and update_tag repeatedly to ensure every file has accurate metadata.

Content Creator

Needs to process new batches of media rapidly. They might use upload_asset for new footage or list_persons to verify if a subject needs a new record, making the workflow faster than manual sorting.

System Administrator

Manages user access and library integrity. They can use list_users and get_user to check who has what permissions or run delete_user if an account needs to be deactivated immediately.

What Changes When You Connect

01 Stop manually sorting photos. Use list_persons and get_person to query the library by face recognition, instantly finding specific memories without keyword guessing.

02 Never lose track of a file again. You can use list_assets or list_albums to see your entire inventory at a glance, then pinpoint details with get_asset when you need it.

03 Maintain clean user data effortlessly. When an employee leaves, instead of manually finding and deleting records, run delete_user and delete_person to scrub the account entirely.

04 Keep your collections organized without clicking through menus. Simply ask your AI client to create_album a new grouping based on dates or events, then populate it automatically.

05 Speed up content ingestion. Instead of zipping files and uploading them via the UI, use upload_asset to push large batches of media directly from your local folder into Immich.

06 Monitor system health quickly. Use get_server_info to instantly verify that everything is running smoothly before a major project rollout.

Real-World Applications

Auditing old event photos

A corporate historian needs to prove who was at the 2019 conference. They ask their agent: 'List all people from the 2019 conference and get details for John Smith.' The MCP uses list_persons, then gets_person, delivering a structured report instead of forcing them into the web UI.

Cleaning up user data

An HR manager needs to decommission an old account. They tell their agent: 'Deactivate User Jane Doe.' The MCP runs delete_user, then confirms that all associated tags and albums are managed or archived.

Batch uploading new content

A photographer just finished a shoot and has 500 RAW files. Instead of using the standard upload process, they instruct their agent to 'Upload these 500 assets.' The MCP runs upload_asset for all files, getting them into the system instantly.

Creating a retrospective gallery

The family wants to compile photos from multiple years. They ask: 'Create an album called Family Reunion 2025 and add any assets tagged with 'Grandma'.' The MCP runs create_album, then uses list_tags/get_tag logic to populate it.

Patterns to Avoid

Trying to find a file manually

✗ AVOID

The user opens the Immich web interface and clicks through 15 nested folders, using keyword searches across various tabs until they locate one specific photo.

✓ INSTEAD

Instead, ask your agent to get_asset using the asset ID or simply prompt: 'Find me the picture of Sarah with the yellow dog.' The MCP handles the lookup automatically.

Deleting data in stages

✗ AVOID

The user manually deletes a person's profile, then separately delete all associated tags, and finally try to list assets. They forget one step.

✓ INSTEAD

Use the MCP's comprehensive tools: run delete_person first, then instruct your agent to use delete_tag on all related tags, ensuring no orphaned data remains.

Ignoring user permissions

✗ AVOID

A new team member manually tries to create a user account without checking if the system allows it or updating their existing profile details.

✓ INSTEAD

Always use get_user first to verify the current status, then run update_user with specific fields before creating any accounts using create_user.

The Right Fit

Use this MCP if your core problem is managing metadata and structured data *within* an existing self-hosted media library. You need conversational control over CRUD operations like updating assets, managing albums, or listing people by identity. Don't use it if you primarily need a simple cloud backup solution; for that, a dedicated syncing service is better. Also, don't use this if your goal is just to view photos—you need the MCP because you want to

change things (add tags, delete users, create albums) via natural language commands instead of clicking through forms.

Handling a massive digital photo archive is always a nightmare.

Today, managing your media means jumping between the web interface, administrative panels, and maybe even a spreadsheet just to keep track of who owns what tag or which albums are complete. You click into 'Users,' then you find the list; you copy an ID number; you switch tabs to 'Albums' to see if that user has any related content; it’s slow, tedious clicking.

With this MCP, that multi-step process disappears. Your AI client talks directly to Immich. You don't click through user roles or folder trees; you just tell your agent what you need done—like getting details for a specific person using get_person—and it handles the entire chain of commands.

Control Your Library Structure with Immich MCP

The biggest manual time sinks are creating new albums and updating metadata. Instead of building an album by manually selecting dozens of photos, you can ask the agent to create_album a grouping based on a date range, or use update_asset to mark 50 files as favorites in one go.

What's different now is that your AI client doesn't just retrieve information; it actively modifies the structure. You're moving from viewing data to architecting your entire media library using conversational commands.

Immich (Google Photos Alternative) with 26 Tools

These tools allow you to perform every administrative and organizational task on your Immich instance—from creating new tags to deleting entire user accounts.

#	TOOL	DESCRIPTION
01	`create_album`	Creates a brand new album within your Immich photo library.
02	`create_person`	Adds a new identity record for a person found in the photos.
03	`create_tag`	Establishes a new descriptive tag you can apply to various assets.
04	`create_user`	Sets up an account for a new user within the Immich system.
05	`delete_album`	Removes an entire album and all its associated contents from the library.
06	`delete_asset`	Permanently deletes a specific photo or video file from the collection.
07	`delete_person`	Removes an identity record for a person, cleaning up obsolete data.
08	`delete_tag`	Deactivates or removes a tag that was previously used across the library.
09	`delete_user`	Removes an existing user account from the Immich system.
10	`get_album`	Retrieves all details about a specific, named album.
11	`get_asset`	Fetches metadata and information for one unique photo or video file.
12	`get_person`	Retrieves the profile details for a recognized person in your photos.
13	`get_tag`	Gets all information about an existing tag, including its usage count.
14	`get_user`	Retrieves the current profile and permissions for a specific user account.
15	`list_albums`	Generates a comprehensive list of every album currently stored in Immich.
16	`list_assets`	Outputs a list containing the titles and IDs of all photos and videos available.
17	`list_persons`	Returns a full directory listing of every person face recognized by the system.
18	`list_tags`	Outputs all tags that have been created and used across your entire media collection.
19	`list_users`	Provides a complete list of every user account connected to Immich.

#	TOOL	DESCRIPTION
20	get_server_info	Checks the current operating version and overall status of the Immich instance.
21	update_album	Modifies the details or contents of an existing album without deleting it.
22	update_asset	Changes metadata, like marking as a favorite, on an already uploaded photo or video.
23	update_person	Modifies a person's record, such as correcting their name or adding context.
24	update_tag	Changes the definition or scope of an existing tag across the library.
25	update_user	Modifies a user's profile details or permissions level.
26	upload_asset	Transfers and adds a new photo or video file directly into the Immich library from your local machine.

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 photos in Immich.

I've retrieved your assets. You have 150 photos and videos available. Would you like to see the details of the most recent one?

U Create a new album called 'Summer Vacation 2024'.

The album 'Summer Vacation 2024' has been successfully created with UUID: 8f2d... You can now start adding assets to it.

U Get the details for asset 550e8400-e29b-41d4-a716-446655440000.

Inspecting asset... This is a JPEG image taken on 2023-08-15. It is currently marked as a favorite and is part of the 'Travel' album.

Frequently Asked Questions

01 **How do I use Immich MCP to upload photos?**

You use the upload_asset tool. Simply ask your agent to run this tool and specify the local folder path or file names you want added to your library.

02 **Can I list all users with Immich MCP?**

Yes, use the list_users tool. This provides a complete directory of every account currently connected to your Immich instance for quick auditing or status checks.

03 **What if I need to correct someone’s name? Should I use update_person?**

Yes, you run the update_person tool. This allows you to modify details like a person's recognized name or add descriptive context without deleting their profile entirely.

04 **How do I check if my Immich server is running okay?**

You use get_server_info. This single tool gives you the current operational status and version number, letting you verify system health instantly.

05 **Does Immich MCP let me delete assets? Is it safe?**

Yes, the delete_asset tool permanently removes files. Since this is a powerful action, your AI client will prompt for confirmation before running any deletion command to prevent accidental loss.

06 **Can I list and update tags across my library?**

Absolutely. You can use list_tags to see what exists, then use get_tag to check details, and finally run update_tag if you need to change the tag's scope or definition.

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 `"immich-google-photos-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

Immich (Google Photos Alternative) 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 →

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Immich (Google Photos Alternative). 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	Immich (Google Photos Alternative) MCP
Server ID	`019e38ac-d696-714b-b6c0-37473a9dc532`
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/immich-google-photos-alternative.

© 2026 Vinkius · vinkius.com · support@vinkius.com