MCP SERVER | NO CODE | CLOUD HOSTED

Aliyun OSS MCP for AI Agents

Manage Cloud Assets, Buckets, and Metadata Storage

Aliyun OSS MCP connects your AI agent directly to Aliyun Object Storage, China's top cloud asset management platform. Use it to programmatically handle large-scale file operations—from uploading text assets and auditing metadata to listing bucket contents with advanced filters—all through natural conversation.

A+ Quality Score 100/100

object-storage | file-management | metadata-auditing | cloud-infrastructure | bucket-management | data-archiving

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

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.

Aliyun OSS / 阿里云对象存储 MCP

10 tools available
Cloud-hosted on Vinkius

Managing vast amounts of cloud storage data usually means wrestling with complex web consoles or writing brittle API scripts. This MCP changes that. It lets your AI client treat Aliyun OSS like a natural extension of your workflow. Need to check the access control list for an entire bucket? Or maybe you just need to verify which objects fall under a specific folder prefix? Instead of clicking through menus, you talk to your agent.

With this connection, your agent becomes a real-time cloud storage assistant. You can upload new content, retrieve detailed object metadata (like HTTP headers), or even automatically generate public URLs for shared assets—all without leaving your AI chat window. It's about moving past the clicks and into conversation. By integrating Aliyun OSS via Vinkius, you get instant access to a powerful suite of cloud tools, letting developers and content operations staff audit files and manage assets using nothing but plain language.

Core Capabilities

01 — Upload and Manage Cloud Objects

Send text content directly to the bucket or copy existing objects within Aliyun OSS.

02 — Audit Bucket Metadata and ACLs

Retrieve detailed configuration, storage statistics, and access control lists for any given bucket.

03 — Discover and Filter Storage Contents

List all objects in a bucket using prefixes or markers to narrow down searches efficiently.

04 — Read and Write File Data

Download text-based files, ensuring compatibility with JSON structures, or upload new content by text.

05 — Secure Asset Lifecycle Management

Generate public URLs for sharing assets or delete objects entirely when they are retired.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and supply your Aliyun AccessKey ID, Secret, Endpoint, and the specific Bucket Name.

02 Your AI client authorizes the connection, allowing it to interact with your cloud storage using natural language prompts.

03 You tell your agent what you need—for instance, 'List all objects in the marketing folder'—and the agent performs the necessary operations through the connected tools.

The bottom line is that you get a conversational API layer over complex cloud infrastructure, allowing your AI to execute storage commands directly.

Built For

This MCP is built for technical roles who spend time coordinating large amounts of digital content or managing infrastructure. Think DevOps engineers tired of writing boilerplate deployment scripts, or Content Operations specialists who need to audit metadata across hundreds of files manually.

DevOps Engineer

Automating asset deployments and monitoring storage configurations via natural language queries instead of running multiple command-line tools.

Content Operations Specialist

Coordinating content refreshes, checking file permissions (ACLs), and auditing media metadata directly from their AI workspace.

Software Developer

Integrating professional cloud storage APIs into daily development routines to manage temporary files or configuration data without leaving the chat interface.

What Changes When You Connect

01 Audit entire asset repositories instantly. Instead of running separate reports on file metadata, use the `get_object_metadata` tool to retrieve detailed HTTP headers for any object in minutes.

02 Automate content deployment pipelines. Use `upload_object` or `copy_object` to move and save new text assets without manually interacting with a console interface.

03 Gain full visibility into your storage footprint. Quickly check the overall health using `get_bucket_statistics`, knowing exactly how much space you're consuming across all buckets.

04 Simplify sharing files. If you need an external link, your agent can automatically generate public endpoints for shared assets after confirming necessary permissions with `get_bucket_acl`.

05 Streamline asset discovery. The `list_objects` tool lets you filter massive directories by path or use pagination markers to pinpoint exactly what you're looking for.

Real-World Applications

Checking Compliance Before a Major Audit

A compliance officer needs to confirm that all sensitive documents in the 'legal/archive/' folder have been properly tagged and archived. The agent uses `list_objects` with prefix filtering, then runs `get_object_metadata` on every result to build a complete audit report.

Migrating Configuration Files

A developer needs to update 50 configuration files across different environments. They use the agent to download all existing content using `download_object_text`, modify it, and then re-upload the updated versions using `upload_object`.

Debugging Broken Links

A team member finds a broken link pointing to an old asset. They ask the agent what the public URL for 'assets/old_image.png' is, and it uses the correct endpoint generation logic to fix the reference.

Validating Bucket Permissions

The DevOps team needs to ensure that only authorized services can read from the main data bucket. They prompt the agent to run `get_bucket_acl` and receive an immediate, accurate report on current permissions.

Patterns to Avoid

Treating OSS like a simple file share

✗ AVOID

Assuming you can just 'write' to the bucket without worrying about versioning or ACLs. This leads to accidental overwrites and data loss because permissions aren't checked.

✓ INSTEAD

Always confirm permissions first using `get_bucket_acl`. If you need to make sure a file is safe, use `copy_object` to create a duplicate in a secure staging area before modifying the original.

Listing objects without filtering

✗ AVOID

Asking the agent to list all objects in a massive bucket. This overwhelms the system and only returns thousands of irrelevant filenames, making it impossible to find what you need.

✓ INSTEAD

Always use the prefix argument with `list_objects`. Specify the path (e.g., 'images/') so the agent narrows the search down immediately.

Ignoring file type requirements

✗ AVOID

Trying to download a binary video file using text-only commands, resulting in corrupted or unreadable data.

✓ INSTEAD

If you need content for processing by an agent, use `download_object_text` if it's JSON/text. If the goal is just visibility, always start with `get_object_metadata` to confirm the file type.

The Right Fit

Use this MCP when your workflow requires managing cloud assets programmatically and at scale. Specifically, use it if you need to audit metadata (`get_object_metadata`), manage access controls (`get_bucket_acl`), or execute bulk operations like copying or deleting objects. Don't use this if your primary task is simply accessing a single file via a known URL; in that case, standard

networking tools suffice. You also don't need it if you only interact with one specific cloud provider (e.g., AWS S3) and never deal with the regional nuances of Aliyun OSS.

Aliyun OSS MCP for Automating Cloud Asset Deployments

Currently, deploying a new set of assets requires multiple manual steps. You have to log into the console, navigate through folders, and either manually upload each file or run complex scripts that handle naming conventions, versioning, and metadata tagging. It's tedious, time-consuming, and prone to human error.

With this MCP, you simply tell your agent: 'Deploy the Q3 assets.' The agent handles the entire process—uploading all required text content using `upload_object`, verifying the folder structure with `list_objects`, and setting up the correct access controls. You get a confirmation that everything is live and correctly placed.

Aliyun OSS MCP for Comprehensive Metadata Auditing

Before, auditing assets meant writing custom code to hit separate endpoints just to check the HTTP headers. You'd have to stitch together information about file size, last modified date, and custom tags from multiple sources.

Now, you ask your agent for a full audit report on the 'legal/archive/' bucket. The agent uses `get_object_metadata` across all files, compiling the exact details into one readable summary. You get instant, deep insights without writing boilerplate code.

Aliyun OSS / 阿里云对象存储 MCP: 10 Cloud File Management Tools

These tools let your AI agent perform specific cloud actions like copying files, downloading text data, listing objects by prefix, and checking bucket status.

#	TOOL	DESCRIPTION
01	`copy_object`	Copies an object from one location to another within your OSS bucket using specific headers.
02	`delete_object`	Permanently removes a specified object file from the Aliyun OSS storage.
03	`download_object_text`	Retrieves an uploaded object, specifically optimized for downloading text or JSON content formats.
04	`get_bucket_acl`	Checks and returns the current access control list defining who has permission to interact with a bucket.
05	`get_bucket_info`	Retrieves general configuration details about a specific OSS bucket.
06	`get_bucket_location`	Identifies the geographic region where your specified storage bucket is physically located.
07	`get_bucket_statistics`	Calculates and returns aggregated data regarding total storage usage for a given bucket.
08	`get_object_metadata`	Fetches the detailed metadata (HTTP headers) associated with any single stored object file.
09	`list_objects`	Provides a list of all objects in the bucket, allowing filtering by path or paginating results using markers.
10	`upload_object`	Transmits and saves raw text content up to 5GB into your specified OSS bucket location.

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 old profile pictures in the 'avatars/' folder and tell me their metadata.

Bucket: user-assets

Found 45 items under `avatars/`.

KEY	SIZE	LAST MODIFIED	STATUS
avatars/jdoe.png	120 KB	2023-01-15	
avatars/asmith.jpg	89 KB	2023-05-22	
... (3 items) ...			

Do you want me to retrieve the detailed metadata for `avatars/jdoe.png`?

U I need to upload this new policy document and make sure it's public.

Done! The text content was successfully uploaded to `policies/2024/policy_v3.txt`.

To confirm the sharing status, I ran a check on permissions:

- **Bucket:** user-assets
- **ACL Status:** Public Read Access Granted ✅

The document is now available at: `https://[bucket].[endpoint].aliyuncs.com/policies/2024/policy_v3.txt`

U What's the current status and location of my primary data bucket?

Bucket Location: The `primary-data` bucket is hosted in the **China (Hangzhou)** region.

Storage Statistics:

- Total Objects: 1,450,220
- Current Storage Used: 7.8 TB
- Last Updated: Just now.

Frequently Asked Questions

01 **How do I manage my cloud files using the Aliyun OSS MCP for AI Agents?**

You talk to your agent as if it were a colleague. Instead of navigating menus, you just tell it what to do—like listing contents or checking metadata. Your agent handles all the complex API calls behind the scenes.

02 **Can this MCP handle large amounts of data uploads?**

Yes, you can upload up to 5GB per request. It's designed for operational tasks that involve moving and saving substantial text-based assets into your cloud storage.

03 **Do I need to know AWS or Azure commands to use the Aliyun OSS MCP?**

No. This MCP abstracts away all the specific technical jargon. You interact using plain English prompts, and the agent translates those instructions into precise cloud storage commands.

04 **How do I check if a file is publicly viewable with this MCP?**

You can ask your agent to retrieve the bucket's access control list (`get_bucket_acl`) or specifically request the public URL. It confirms exactly what permissions are set, preventing accidental link sharing.

05 **What kind of data is best for this Aliyun OSS MCP?**

It excels with text-based content like JSON, configuration files, and documentation. If you need to download the file contents as plain text, it handles that process efficiently.

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 `"aliyun-oss": { "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

Aliyun OSS / 阿里云对象存储 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

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Aliyun OSS / 阿里云对象存储 MCP
Server ID	`019d8415-2362-7211-bf87-43552a7fac47`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE