MCP SERVER | NO CODE | CLOUD HOSTED

Tencent COS MCP

Manage files, metadata, and asset lifecycles.

Tencent COS / 腾讯云对象存储 connects your AI agent to China's leading cloud storage platform. It lets you manage massive file archives and digital assets using natural language commands. You can upload new content, check if files exist before deployment, get detailed metadata reports, and list directory contents—all without ever logging into the Tencent Cloud Console.

A+ Quality Score 100/100

object-storage | cdn | file-management | data-lifecycle | cloud-assets | metadata-management

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

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.

Tencent COS / 腾讯云对象存储 MCP

10 tools available
Cloud-hosted on Vinkius

This connector gives your AI agent control over complex cloud storage operations, treating your entire asset library like an extension of your chat window. You can instantly upload text assets or delete old files across multiple buckets using simple prompts. Need to audit a directory? Your agent handles listing contents and checking object headers so you know exactly what's in place. It even generates public access points for shared content on demand. Because this MCP is hosted on Vinkius, your AI client gains instant access to dozens of services—making it the single source for orchestrating both your cloud storage and other business processes. You just tell your agent what you need done with your files, and it handles the infrastructure calls.

Core Capabilities

01 — Check File Existence

Quickly verifies if a specific file or object key is present in the bucket.

02 — Get Object Details

Retrieves technical metadata, like content type and storage class, for any given asset.

03 — List Folder Contents

Displays the full list of files or subdirectories within a specified path using advanced filters.

04 — Upload and Modify Assets

Sends text content to the cloud, uploads new objects, or copies existing ones within the storage structure.

05 — Download Text Content

Pulls down the raw text data from a specified object for immediate use by your agent.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and input your specific Tencent Cloud credentials, including the SecretId, SecretKey, Region, and target Bucket Name.

02 Connect your AI client (like Claude or Cursor) to Vinkius, giving it permission to use this storage tool.

03 Ask your agent a question—for example, 'List all files in the root folder'—and get an actionable list of results back.

The bottom line is that you interact with cloud storage like talking to a colleague, not navigating a complex web console.

Built For

This MCP targets operations engineers and content managers who waste time clicking through dashboards just to track down a file version or check permissions. It's for anyone whose job requires interacting with cloud assets without needing deep API knowledge.

DevOps Engineer

Automates asset deployment, ensuring new content is properly uploaded and that old resources are deleted safely.

Content Manager

Coordinates content refreshes by getting object metadata and generating public URLs for marketing assets on the fly.

Auditor

Runs deep audits by checking object existence or listing entire directory structures to verify compliance.

What Changes When You Connect

01 You gain immediate visibility into your entire storage setup. Instead of clicking through multiple tabs to check permissions, you just ask the agent to get bucket access permissions.

02 Asset management becomes instant. Need to move a file? Use the copy_object tool to duplicate content without manual steps, and use delete_object when an asset is retired.

03 Audit files with precision. The list_objects tool lets you filter by path or delimiter, organizing your storage structure in a single request rather than piecing together results.

04 Metadata queries are simple. Use get_object_metadata to grab technical details like the content type or ETag for any file so your agent can process it correctly.

05 Streamline content updates. You can upload_object text assets and instantly generate public endpoints, making them ready for use without console navigation.

Real-World Applications

Auditing a client's document archive

An operations manager needs to know if 'ClientX/Q3/final.pdf' exists and what its last modified date was. Instead of manually checking the file path, they ask their agent to check_object_exists and get_object_metadata, getting a precise answer instantly.

Preparing content for public release

A marketing team needs to make five new images available. They ask their agent to upload_object the text descriptions and generate public endpoints automatically, making them accessible without manual URL setup.

Verifying data integrity before migration

A developer must confirm that a critical configuration file is still present before deploying code. They use head_bucket first to verify the entire bucket connection, then check_object_exists for the specific key.

Analyzing old data structures

An auditor needs to see every folder and subfolder under 'archive/'. Instead of traversing deep directory trees, they use list_objects with advanced filters to get a complete map.

Patterns to Avoid

Listing all files manually

✗ AVOID

Trying to browse through the Tencent Cloud Console folder by folder and copy-pasting filenames into a spreadsheet. This is slow, error-prone, and impossible for deep archives.

✓ INSTEAD

Use list_objects with prefix filtering to get an organized inventory in one shot. If you need just the top level, use list_root_objects.

Checking file status via API calls

✗ AVOID

Writing complex code that makes multiple sequential API calls (HEAD, GET) just to see if a single asset exists and what its type is.

✓ INSTEAD

Use get_object_metadata first. It gives you the necessary technical details in one function call, telling you exactly what you need.

Updating an asset's location

✗ AVOID

Downloading a file, renaming it locally, and then re-uploading it to COS because the path changed.

✓ INSTEAD

Use copy_object. This copies the file from one place to another within the cloud without forcing a full data transfer or loss of metadata.

The Right Fit

You should use this MCP if your workflow involves reading, writing, organizing, or auditing files stored in Tencent COS using natural language prompts. It's perfect for operations teams who need to confirm file existence (check_object_exists), list large directories (list_objects), or generate public links from text content (upload_object). Don't use this if your primary goal is database management, user authentication, or complex business logic that doesn't involve files. For pure data transformation tasks, you might need a different MCP; but for anything involving the file system itself, this connector provides the necessary control.

Dealing with Cloud Storage Means Jumping Through Too Many Rings.

Today, managing cloud assets means logging into the Tencent Console. You have to navigate menus just to check if a file exists or what its permissions are. If you need to list 50 directories across different folders, it's a tedious process of clicking 'list contents,' then copying that list, and pasting it somewhere else for review.

With this MCP, your AI agent handles the complexity behind the scenes. You simply ask: 'What files do I have in my backups folder?' The answer comes back immediately, complete with metadata details, without you ever having to click a single button on the Tencent Cloud Console.

Tencent COS / 腾讯云对象存储 Provides Deep Asset Control.

Gone are the days of manual file verification. You no longer have to write multi-step scripts just to check if a bucket is configured correctly, or run separate commands to list objects versus checking metadata.

Now you get direct, conversational control over your storage infrastructure. It's immediate, accurate cloud coordination that happens inside your chat window.

Tencent COS / 腾讯云对象存储 with 7 Tools

These tools allow you to perform specific actions like checking existence, copying objects, or listing directory contents directly through your AI agent.

#	TOOL	DESCRIPTION
01	`check_object_exists`	Confirms whether a specific file key is present in the bucket.
02	`copy_object`	Duplicates an existing object from one location to another within your bucket.
03	`delete_object`	Permanently removes a specified file from the COS storage.
04	`download_object_text`	Downloads the text content of an object and presents it to your agent for use.
05	`get_bucket_acl`	Retrieves the current access permissions settings for the entire bucket.
06	`get_object_metadata`	Fetches technical details about a file, such as its size or content type.
07	`head_bucket`	Checks if the bucket itself is correctly configured and accessible via API.
08	`list_objects`	Use prefix to filter by path. List objects in the COS bucket
09	`list_root_objects`	List top-level objects and folders
10	`upload_object`	Max 5GB per request. Upload text content to COS

See It in Action

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

U List all files in the root of my Tencent COS bucket.

I've retrieved the root object list. I found 10 items, including 'index.html' and 'uploads/'. Would you like to see the contents of a specific folder or get metadata for a file?

U Check if the file 'backups/db_init.sql' exists in COS.

I've performed a metadata check for that key. The file `backups/db_init.sql` exists in your bucket. It was created on Oct 27, 2023 and has a size of 1.2 MB. Should I generate a public URL for it?

U Get the metadata for 'static/css/main.css'.

I've retrieved the metadata for that object. The Content-Type is 'text/css' and it has an ETag of '88210934...'. The storage class is Standard. Would you like me to download the text content for you to audit?

Frequently Asked Questions

01 **Does Tencent COS / 腾讯云对象存储 support listing all files?**

Yes, it does. You can use list_objects to view the contents of a bucket and even filter by specific paths or folder markers.

02 **How do I check if an object is already in the bucket using Tencent COS / 腾讯云对象存储?**

You run the check_object_exists tool. This sends a quick metadata query and tells you immediately if that specific file key exists, saving time compared to downloading it just to verify.

03 **Can Tencent COS / 腾讯云对象存储 handle large files?**

Yes. The upload_object tool supports up to 5GB per request, giving you plenty of room for large text assets and documents.

04 **What is the difference between list_objects and list_root_objects with Tencent COS / 腾讯云对象存储?**

list_root_objects shows only the top-level contents, like main folders. list_objects lets you specify a prefix to drill down into specific paths deeper in your storage structure.

05 **Do I need special coding knowledge for Tencent COS / 腾讯云对象存储 MCP?**

No. You interact with it using natural language prompts through your AI client, so you don't need to write code or worry about the underlying API calls.

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 `"tencent-cos": { "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

Tencent COS / 腾讯云对象存储 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 →

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 Tencent COS / 腾讯云对象存储. 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	Tencent COS / 腾讯云对象存储 MCP
Server ID	`019d8489-df4c-73ad-9454-ce9b7662cded`
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/tencent-cos.