MCP SERVER | NO CODE | CLOUD HOSTED

Google Cloud Storage Bucket MCP

Securely store and manage files in a single sandbox.

Google Cloud Storage Bucket MCP gives your agent surgical access: it lets your AI client read, write, list, and delete files inside one specific Google Cloud bucket. This isn't general cloud access; it's a secure sandbox for data persistence, allowing your agent to manage work assets, store generated reports, or analyze documents without touching your wider cloud infrastructure.

F Quality Score 3.6/100

object-storage | file-management | data-persistence | cloud-storage | storage-buckets

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/google-cloud-storage-bucket

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.

Google Cloud Storage Bucket MCP

4 tools available
Cloud-hosted on Vinkius

Your AI client needs a place to keep things—a temporary hard drive that doesn't mess with the rest of your production setup. This MCP provides exactly that: highly contained access to one specific Google Cloud Storage Bucket. By limiting permissions so strictly, you give your agent an isolated area where it can safely store data and process information. It’s perfect for agents running complex jobs or managing large document sets. Instead of giving the AI keys to your entire cloud account, this MCP gives it a digital filing cabinet with one lock on it.

Your agent can now upload new configurations, retrieve historical reports, read raw source files, and even delete temporary assets when they're done. This containment is huge. It means you get the power of scalable object storage without introducing global security risk to your core systems. You connect this MCP through Vinkius, treating it just like any other specialized tool in our catalog.

Core Capabilities

01 — List existing files

The agent can list all the file names and paths within the configured cloud bucket.

02 — Read file content

You instruct the agent to read a specific object, returning its full text or binary contents.

03 — Upload or overwrite files

The agent can upload new data or replace existing objects within the bucket.

04 — Delete assets

You ask the agent to permanently remove a specified file from the cloud storage bucket.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/google-cloud-storage-bucket** — connect your AI agent in three steps.

01 First, you connect your AI client through Vinkius and point it to this MCP. You define exactly which Google Cloud Storage Bucket the agent is allowed to touch.

02 Next, you tell your agent what needs doing—for example, 'List all CSV files in the data/exports folder.' The agent then executes the necessary tool call against the scoped bucket.

03 Finally, the system returns a clean list of file names or the requested content. You use that information to continue your workflow.

The bottom line is you get secure, focused access to cloud storage without exposing your agent to dangerous global permissions.

Built For

Backend engineers and data ops managers need this. If your workflow involves an AI agent needing a reliable, dedicated place to store temporary results, uploaded configurations, or historical records, you're in the right spot. It solves the problem of giving an agent 'just enough' access.

Data Scientist

They need to save large processed datasets and configuration files that their AI model generates for later retrieval.

Backend Engineer

They use it when building agents that manage external resources, requiring the agent to read or write structured data like user profiles or job logs.

DevOps Operator

They want an automated way for AI systems to handle temporary build artifacts and deployment files in a quarantined location.

What Changes When You Connect

01 Absolute Security: Because the agent is locked to one bucket, you prevent it from listing or touching other critical company data outside of this dedicated area. You control exactly what it sees.

02 Data Persistence: Need your AI client to remember something? Use `put_object` to upload generated assets, reports, and configuration files for later retrieval by the agent itself.

03 Content Analysis: If you need your agent to analyze a raw data file, use `get_object` to read its contents directly into the prompt context without manual downloads or uploads.

04 Workflow Cleanup: Use `delete_object` to automatically clear out temporary files (like job logs or cached exports) once they are no longer needed. This keeps your storage tidy.

05 Visibility: The `list_objects` tool gives you a quick, programmatic inventory of all the current assets in the bucket, perfect for auditing purposes.

Real-World Applications

Archiving Model Outputs

A data scientist runs an intensive simulation. Instead of emailing hundreds of CSVs, they prompt their agent to `put_object` all the results into the bucket. Later, the agent can use `list_objects` and `get_object` to gather a comprehensive report for review.

Processing User Uploads

A web application allows users to upload documents. The system uses the MCP to receive the file via `put_object`. An agent then reads the contents using `get_object` to extract key data points before saving the processed summary.

Cleaning Up Temporary Jobs

A batch processing job finishes and leaves behind large temporary files. The engineer instructs the agent to use `delete_object` on all known temp paths, ensuring no junk data remains in the bucket.

Patterns to Avoid

Assuming global access

✗ AVOID

Trying to give your AI client full GCP credentials so it can 'find' a file somewhere on the network.

✓ INSTEAD

Don't use general cloud connections. Use this MCP because its scoping forces the agent to stay within one dedicated bucket, making operations safer and more predictable.

Manually managing files

✗ AVOID

A developer has to write a script that connects to GCP, lists files via CLI, reads content into Python variables, processes them, and then uploads the result.

✓ INSTEAD

Let your agent handle it. Use `list_objects` first, pass the resulting filenames to an action sequence, and let the agent use `get_object`, process the data, and finally save the output using `put_object`.

Over-engineering file handling

✗ AVOID

Writing complex code just to check if a file exists before reading it.

✓ INSTEAD

If you need to know what's there, use `list_objects`. If you want to read it, the agent handles the logic of attempting to `get_object` safely.

The Right Fit

Use this MCP if your primary need is managing files and data persistence within a single, isolated storage location. You need an agent to act like a digital file manager: listing contents, reading specific files, uploading new versions, or deleting junk artifacts. It's ideal for sandboxed tasks, such as processing user uploads, storing temporary model results, or holding configuration assets.

Don't use this if you need the AI client to interact with other cloud services (like databases, message queues, or compute instances). If your goal is to perform actions outside of file CRUD operations—for example, sending an email or calling a third-party API—you need a different specialized MCP. This tool only manages objects inside one bucket.

The headache of managing temporary files in the cloud

Today, when your agent finishes a large job, you're left with dozens of output files: raw data dumps, processed exports, logs, and intermediate JSONs. You have to write complex code just to list all those files, manually pass them through several functions, read the content into memory, and then decide which ones are safe to delete or which need to be saved for audit.

With this MCP, your agent handles it in a single conversation flow. It uses `list_objects` to see what's there, reads only the files you specify with `get_object`, processes them, and then saves the clean result using `put_object`. The whole process stays contained.

Accessing Data Objects with Google Cloud Storage Bucket MCP

You no longer have to write boilerplate code just for basic file operations. Instead of writing a multi-step script that connects, lists, reads, and writes, you simply tell your agent the goal.

The difference is control. You get specific, scoped superpowers—you can read content with `get_object` or upload new versions using `put_object`, all without exposing unnecessary permissions to the broader cloud environment.

Google Cloud Storage Bucket: 4 Tools

Use these four tools to programmatically list, retrieve, upload, and delete specific file assets within your defined Google Cloud Storage Bucket.

#	TOOL	DESCRIPTION
01	`delete_object`	Removes a specific file object from the Google Cloud Storage bucket.
02	`get_object`	Reads and retrieves the content of a specified file within the cloud storage bucket.
03	`list_objects`	Retrieves a list of all files stored in the configured Google Cloud Storage bucket.
04	`put_object`	Uploads data to the cloud storage, creating a new object or overwriting an existing one.

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 inside the 'data/exports/' folder.

I found 3 files with the prefix 'data/exports/': export-1.csv, export-2.csv, and summary.json.

U Upload this JSON configuration to 'configs/agent-settings.json'.

I've successfully uploaded the JSON data to 'configs/agent-settings.json' with `content-type: application/json`.

U Delete the temporary 'processing/job-123.tmp' file.

The file 'processing/job-123.tmp' was successfully deleted from the Google Cloud Storage bucket.

Frequently Asked Questions

01 **Can I use Google Cloud Storage Bucket MCP to access multiple buckets?**

No. This MCP is intentionally scoped and only grants access to a single, specific bucket. It cannot list or interact with any other storage locations in your cloud account.

02 **How do I upload data using Google Cloud Storage Bucket MCP?**

You use the `put_object` tool. This allows you to either create a brand new file object or overwrite an existing one with updated content.

03 Is deleting objects safe? What does delete_object do?

The `delete_object` tool permanently removes the specified file from the bucket. This is useful for cleaning up temporary files once their job is done.

04 Does Google Cloud Storage Bucket MCP only work with text files?

No. The MCP handles general objects, meaning you can read and write various types of data, including JSON, CSV, images, or other binary formats.

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 `"google-cloud-storage-bucket": { "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

Google Cloud Storage Bucket 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 Google Cloud Storage Bucket. 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	Google Cloud Storage Bucket MCP
Server ID	`019e38a1-d9b4-7276-b89d-b51c929ef3b7`
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/google-cloud-storage-bucket.