MCP SERVER | NO CODE | CLOUD HOSTED

Azure Blob Container MCP for AI Agents

Manage secure cloud data persistence and object storage files

The Azure Blob Container MCP lets your AI securely manage files inside one specific cloud storage area. It provides controlled, high-performance access for reading, writing, listing, and deleting assets. If your agent needs a safe place to persist data or analyze documents without touching global infrastructure, this is it.

A+ Quality Score 100/100

object-storage | file-management | cloud-security | data-persistence | scoped-access | blob-storage

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/azure-blob-container

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.

Azure Blob Container MCP

4 tools available
Cloud-hosted on Vinkius

Most cloud integrations give agents way too much power—global read/write permissions that are huge security risks. This MCP fixes that by giving your AI one surgical superpower: total access only to files inside a single Azure Blob Container. You can trust that the agent stays locked down; it cannot see or touch any other containers or critical backups.

This means you can safely let your AI persist data, process reports, and manage its own working assets without exposing your whole cloud environment. Whether the task is analyzing uploaded documents or simply building a temporary memory cache, this MCP gives your agent a dedicated, private hard drive to work with. It's exactly what you need for secure, contained file operations, connecting it easily via Vinkius's catalog of compatible AI services.

Core Capabilities

01 — Download and read specific files

Your agent can pull down the contents of any target file within the container.

02 — Find all stored assets

It lists every blob (file) in the container, and you can narrow that search down using a specific folder path or prefix.

03 — Create or update files

The agent can write new data to a file or overwrite an existing one with fresh content.

04 — Remove old data safely

You can instruct the agent to delete specific, unnecessary files from the container.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/azure-blob-container** — connect your AI agent in three steps.

01 First, you tell your AI client which operation it needs—for example, 'Find all invoices from last month.'

02 The MCP executes that request by using its scoped permissions to interact with the container and return a list of matching file names or metadata.

03 Finally, your agent receives the actionable data (like a list of files or the actual content) and uses it for its next step, like passing it to another tool.

The bottom line is, you get secure, targeted cloud storage access without having to manage complex permissions across multiple services.

Built For

This MCP targets operations engineers and data scientists who need their AI clients to handle file-based workflows securely. If your team spends time manually zipping up files, checking folder structures, or writing complex, multi-permission cloud scripts, this saves you hours of risk management.

Data Engineer

They use this MCP when building ETL pipelines that require an AI agent to read source data files from a restricted location and then write processed results back into the container.

DevOps Specialist

This role relies on it for automated testing, letting their agents generate configuration files (like YAML or JSON) and safely commit them to a single staging blob container before deployment.

What Changes When You Connect

01 Absolute Security: The agent is strictly locked to one container. You eliminate the risk of accidental deletion or access outside that specific folder.

02 Targeted File Operations: Use `get_blob` to download file contents directly, letting your AI analyze documents without manual downloads and uploads.

03 Structured Workflows: If you need to track assets, use `list_blobs` with a prefix filter. This allows the agent to process only files within an 'invoices/' folder, for example.

04 Reliable Data Persistence: Use `put_blob` when your AI generates reports or summaries; it writes them reliably back into the cloud storage container.

05 Efficient Cleanup: The `delete_blob` tool lets your agent automatically purge old temporary data once a workflow is complete. This keeps the container clean and manageable.

Real-World Applications

Analyzing customer-submitted forms

A marketing analyst needs to review all JSON files uploaded last week. They prompt their agent, which uses `list_blobs` (filtering by 'submissions/'), then calls `get_blob` on each file to pull the raw data into a summary report.

Automating research document storage

A legal team has several documents generated daily. They use their agent to write new, timestamped PDFs using `put_blob` and automatically tag them by date, ensuring every file is correctly archived in the container.

Cleaning up temporary assets

After a major data processing run, thousands of temporary CSV files are created. The agent uses `list_blobs` to find all files matching '*temp*' and then executes `delete_blob` on them in batches, keeping the container clean.

Building an evidence repository

A compliance officer needs to quickly verify if a specific policy document exists. The agent uses `list_blobs`, searching for 'policy_v3.pdf', and confirms its presence or absence, giving instant verification.

Patterns to Avoid

Assuming global access

✗ AVOID

Telling your agent to 'write all documents' without scope limits means it could accidentally modify mission-critical backups stored in a different cloud service.

✓ INSTEAD

Always use the contained power of this MCP. If you need to write data, use `put_blob` only after confirming the target file is within the designated container.

Overcomplicating retrieval

✗ AVOID

Trying to manually download 50 files and then re-uploading them into a structured database just to read their contents.

✓ INSTEAD

Use `get_blob` directly. It pulls the file content straight to your agent for immediate processing, skipping the manual transfer steps.

Forgetting prefixes

✗ AVOID

Asking the agent to list files without specifying a folder path causes it to return thousands of irrelevant results, slowing down the entire process.

✓ INSTEAD

When listing files, always provide a prefix (like 'invoices/') so the `list_blobs` tool only returns relevant results.

The Right Fit

Use this MCP if your core need is managing data persistence *within* a single, highly restricted cloud storage location. It's perfect when you need controlled read/write access to files and folders but absolutely cannot risk the agent accessing other parts of your infrastructure (like databases or network resources). Don't use it if you need to connect to multiple distinct buckets or containers; this MCP is scoped to just one. If your goal involves complex data transformation logic that requires a separate service, pair this MCP with an orchestration layer, but keep the file access itself here.

Azure Blob Container MCP for AI Agents: Managing Cloud File Storage

Today, managing cloud assets is a messy process. You're constantly clicking through different cloud consoles, checking permissions across dozens of services just to find the right file. If you need an agent to read a document or save a report, it often requires multiple manual steps: finding the correct bucket, verifying write access, and then initiating the download/upload.

With this MCP, your agent handles all that complexity behind the scenes. You simply ask for the data—'Get me the latest quarterly summary.' The agent uses its contained tools to securely find, read, or save the file in one go. It just works.

Azure Blob Container MCP for AI Agents: Data Lifecycle Management

Data retention is a headache. You have files that are temporary—test logs, generated summaries—that pile up and clutter your storage, costing you money and slowing down searches. Manually tracking which files need to be deleted after a project ends is time-consuming.

Now, you can build automated workflows where the agent runs cleanup jobs. After processing a batch of records, it automatically uses `delete_blob` on the temporary files. It's clean, reliable data lifecycle management.

Azure Blob Container: 4 Tools for File Management Ops

Use these tools to securely list contents, download file data, upload new assets, or delete old temporary files within a single Azure container.

#	TOOL	DESCRIPTION
01	`delete_blob`	Wipes out a specified file from the configured storage container.
02	`get_blob`	Downloads and reads all the content from a specific, targeted file.
03	`list_blobs`	Retrieves a list of files inside the container; you can optionally filter results by folder path.
04	`put_blob`	Creates a new file or updates an existing one with provided content in the container.

See It in Action

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

U List all PDF files in our 'client_reports/' folder that were added last month.

I found 4 reports matching your criteria:

- `client_reports/AcmeCorp_2025-01.pdf` (Size: 3MB)
- `client_reports/BetaSolutions_Q1.pdf` (Size: 7MB)
- `client_reports/GlobalInc_Summary.pdf` (Size: 12MB)
- `client_reports/FinalReview_05-24.pdf` (Size: 2MB)

U Read the contents of 'config.json' and tell me what retention period is set.

Configuration Details

The file `config.json` contains the following settings:

KEY	VALUE
`retentionDays`	30
`active`	true

Retention is set to 30 days.

U Save this summary of findings as 'final/report_v2.txt'.

✅ Success! The file `final/report_v2.txt` has been written successfully to the container. You can now reference it in your workflows.

Frequently Asked Questions

01 **How does the Azure Blob Container MCP keep my data safe from global permissions?**

It keeps your data contained by strictly limiting all actions to one specific storage container. The agent can't see or modify any other cloud resources, making it incredibly secure for sensitive workloads.

02 **Can I use the Azure Blob Container MCP to find files in a specific virtual folder?**

Yes. When listing files, you just provide the 'folder' path (or prefix) in your request. The agent will only return results that match that exact directory structure.

03 **Is this MCP good for temporary data storage and cleanup?**

It’s excellent for that. You can write code that automatically lists all files matching a pattern (like 'temp*') and then uses the delete tool to clean them up, keeping your cloud container tidy.

04 **What kind of data types can I store or retrieve using this MCP?**

It handles any object type—PDFs, JSON files, CSV spreadsheets, images, plain text. As long as it's an object stored in Azure Blob Storage, your agent can manage it.

05 **Does the Azure Blob Container MCP allow me to read private client documents?**

Yes, provided the AI agent has the necessary credentials for that container. It reads the file content directly into your workflow without you needing to manually download and paste it.

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 `"azure-blob-container": { "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

Azure Blob Container 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 Azure Blob Container. 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	Azure Blob Container MCP
Server ID	`019e3869-5853-7159-85f3-56e1a7739626`
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/azure-blob-container.