MCP SERVER | NO CODE | CLOUD HOSTED

Backblaze B2 MCP for AI Agents

Manage cloud object storage and file versions in your development workflow

Backblaze B2 MCP gives your AI agent direct control over cloud object storage management. You can audit buckets, create and delete partitions, manage file versions (hard/soft deletes), and check for failed large file uploads without leaving your editor.

A+ Quality Score 100/100

s3-compatible | object-storage | file-management | data-archiving | bucket-management | cloud-backup

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/backblaze-b2

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.

Backblaze B2 MCP

10 tools available
Cloud-hosted on Vinkius

Managing cloud infrastructure is complex, especially when you're dealing with version history and cleanup jobs. This MCP connects any AI agent to the full power of Backblaze B2 storage management. Instead of navigating multiple dashboards or running shell scripts, you talk to your agent and it handles the heavy lifting.

Need to audit what buckets exist or check if a multipart upload failed? Your agent can run those checks instantly. You can also create new private partitions, manage access rules by updating bucket permissions, and even delete old file versions that are hoarding space. It’s all done through natural conversation. If you use Vinkius for your MCP catalog, this tool gives you a single point of control for object storage architecture, allowing you to keep your workflow entirely within your preferred development environment.

Core Capabilities

01 — Audit and Control Bucket Structure

You can list all existing buckets, create new ones, or irreversibly delete empty buckets to properly manage your cloud storage landscape.

02 — Manage File Lifecycles

Mark files as hidden (soft delete) without losing the data, retrieve detailed file information including checksums, or physically remove specific old versions of files from disk arrays.

03 — Validate Large Transfers and Object Integrity

Check for stalled multipart uploads to find failed large file chunks, or get full metadata details like SHA1 hashes to confirm object integrity.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/backblaze-b2** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Backblaze B2 Application Key ID and key credentials.

02 Your AI client uses these keys to connect directly to the specified regional data ingress endpoints assigned to your billing account.

03 You simply ask your agent to perform a storage operation, like 'Show me all public buckets' or 'Delete file version X,' and it executes the command.

The bottom line is that you get immediate, programmatic access to critical cloud infrastructure functions without switching contexts.

Built For

This MCP targets technical roles who spend time debugging storage architecture and managing data lifecycle. It's essential for DevOps Engineers tired of manually running cleanup scripts, Backend Developers needing secure ways to partition assets, or System Administrators performing granular object audits.

DevOps Engineer

Needs to run quick checks on bucket states, clean up abandoned file versions, and debug failed multipart streams immediately after a deployment.

Backend Developer

Creates secure public or private storage domains directly from the development environment before committing code that relies on object storage.

System Administrator

Audits object footprint, checks file integrity using SHA1 hashes, and performs granular cleanup across multiple cloud partitions.

What Changes When You Connect

01 Instantly debug failed uploads. Instead of manually checking logs, you can use the list_unfinished_large_files tool to scan nodes for stalled multipart upload chunks.

02 Control data retention with precision. Use delete_file_version when you need to remove specific old file versions that are unnecessary and contributing to storage waste.

03 Maintain secure assets without extra clicks. You can hide files using the hide_file tool, ensuring they remain available for lifecycle sweeps but are invisible during normal listings.

04 Perform full infrastructure audits quickly. Running list_buckets gives you a complete, up-to-date view of every bucket in your account at a glance.

05 Verify data integrity with confidence. The get_file_info tool provides granular details and SHA1 hashes, confirming that the object you think you have is actually intact.

Real-World Applications

Cleaning up abandoned logs after project migration

A developer suspects an old log bucket has accumulated junk data. They ask their agent to list all buckets, identify the target, and then run delete_file_version repeatedly until the desired object versions are purged.

Debugging broken upload pipelines

A backend team notices large files aren't appearing in storage. They ask their agent to run list_unfinished_large_files, which identifies the exact source of the stalled chunk uploads that need attention.

Auditing public vs private access policies

A sysadmin needs to verify if a new asset bucket is secure. They use list_buckets first, check its current permissions with update_bucket, and then confirm the correct privacy settings are applied before deployment.

Patterns to Avoid

Assuming visibility is enough

✗ AVOID

A user only runs list_file_names and thinks they've seen every file. They miss files hidden due to lifecycle semantics.

✓ INSTEAD

Don't rely solely on listing names; use the hide_file tool first, then run list_file_names if you need a clean view of active assets.

Deleting buckets without checking contents

✗ AVOID

Attempting to delete a bucket that still holds any files or hidden object versions results in an error and leaves data behind.

✓ INSTEAD

Always confirm the target bucket is empty first, then use list_buckets for verification before running the delete_bucket tool.

Ignoring upload failures

✗ AVOID

A large file transfer fails mid-way, but the developer doesn't know it. The data chunk simply stalls in the background.

✓ INSTEAD

Use list_unfinished_large_files to proactively scan for and identify any incomplete multipart uploads.

The Right Fit

You should use this MCP if your workflow requires programmatic control over object storage, especially when dealing with file versioning, bucket state management, or large data transfers. Use it for tasks like running list_file_names to audit assets or using delete_file_version to reclaim space.

Don't use this if you just need general cloud monitoring. If your goal is simply tracking network throughput metrics without modifying objects, look for a dedicated monitoring MCP. Also, if you only need to read metadata and never modify anything, other data retrieval tools might be sufficient; but if deletion or creation are involved, this MCP handles the full lifecycle.

Backblaze B2 MCP: Managing Cloud Object Storage Architecture

Today, managing cloud storage means jumping between a console GUI and running separate scripts just to audit basic bucket states or find out why a file version is missing. You spend time copy-pasting IDs and manually checking logs for failed multipart uploads, which slows down everything.

With this MCP, you simply ask your agent about the storage structure or specific object details. It runs the necessary checks—whether listing_buckets for an overview or running get_file_info on a single asset—and gives you one clean answer. It's instant infrastructure intelligence.

Backblaze B2 MCP: Auditing Object Integrity and File Versions

Manually tracking object integrity is a nightmare. You have to assume that if the file exists, it's correct, but version control makes this impossible. You might forget which versions are actively needed or overlook orphaned data.

This MCP lets you audit and manage versions with precision. By using delete_file_version, you can surgically remove old chunks of data without affecting active assets. It gives you total confidence in your storage cleanup process.

Backblaze B2 MCP: 10 Storage Management Tools for Cloud Architecture

Use these tools to list, create, update, delete, and audit every aspect of your Backblaze B2 cloud storage infrastructure.

#	TOOL	DESCRIPTION
01	authorize_account	Extracts the specific regional data ingress endpoint URL for your billing account to ensure proper connection routing.
02	create_bucket	Provisions a new, empty logical Storage Bucket partition within your B2 account.
03	delete_bucket	Irreversibly deletes an entirely empty Backblaze Storage Bucket; requires confirming no files or versions remain inside.
04	delete_file_version	Permanently removes a specific, historical chunk of data from the disk arrays to reclaim storage space.
05	get_file_info	Retrieves detailed metadata for a single file, including its checksum and content type.
06	hide_file	Marks an active B2 file as hidden using a null-marker. The data remains untouched but is invisible to standard listing commands.
07	list_buckets	Retrieves the exact list of all storage buckets currently existing globally on your B2 account.
08	list_file_names	Paginates a clean list of primary, surface-level object names within a specific bucket.
09	list_unfinished_large_files	Scans the B2 nodes to identify and report on stalled Multipart Upload chunk aggregates that failed to complete.
10	update_bucket	Modifies the global Access Control List (ACL) privacy settings for a specific bucket, useful mainly for static asset endpoints.

See It in Action

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

U What are the names of all the main asset buckets we have right now?

B2 Storage Audit Report

BUCKET ID	PRIVACY STATUS	OBJECT COUNT
`production-assets`	`allPublic`	1,209,000
`internal-logs`	`allPrivate`	540,000
`archive-dumps`	`allPrivate`	3,100

We have three active storage buckets. Do you want to check for any unfinished large files in the `archive-dumps` bucket?

U I need to remove old versions of the quarterly reports from the finance bucket.

Understood. I've identified 14 specific file versions dating back to Q1 that are no longer needed for compliance. **Action:** `delete_file_version` called.

✅ Success: All 14 specified chunks have been permanently removed from the disk arrays. This action is irreversible and frees up significant space.

U Check the file details and SHA1 for 'user-profile-pic.jpg' in the main bucket.

File Details: user-profile-pic.jpg

- **Content Type:** `image/jpeg`
- **Size:** 345 KB globally
- **Unique ID:** `4_zX...`
- **SHA1 Hash:** `2f8bX...` (Confirmed integrity)

The file is fully accessible. Everything looks good to go.

Frequently Asked Questions

01 **How do I check if my Backblaze B2 account has any broken or stalled large uploads?**

You use the MCP's capability to scan for unfinished multipart uploads. This tells you exactly which file transfers failed midway, allowing you to restart them without manual investigation.

02 **Can I safely delete an old bucket using the Backblaze B2 MCP?**

Yes, but you must ensure the bucket is completely empty first. The tool can list all contents and help confirm that no files or versions are left before running the final deletion command.

03 **What's the difference between soft deleting and hard deleting a file?**

Soft deleting (hiding) keeps the data intact but makes it invisible to most users. Hard deleting removes the actual physical data chunks, freeing up storage space immediately.

04 **Does using the Backblaze B2 MCP mean I need to change my workflow?**

Not really. You just talk to your agent like you normally do; it handles the complex cloud interactions behind the scenes. Your workflow remains focused on development, not infrastructure maintenance.

05 **How can Backblaze B2 MCP help me audit my current object storage rules?**

The MCP lets your agent list and verify all existing buckets and their associated access policies. This helps you confirm that data is stored with the correct privacy settings.

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 `"backblaze-b2": { "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

Backblaze B2 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 Backblaze B2. 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	Backblaze B2 MCP
Server ID	`019d7557-f367-7302-9200-58f220cb6eb8`
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/backblaze-b2.