MCP SERVER | NO CODE | CLOUD HOSTED

Appwrite MCP for AI Agents

Automating Backend Infrastructure Audits and Data Management

Appwrite MCP connects your AI agent directly to an open-source backend service. It lets you manage databases, audit user accounts, and monitor cloud storage resources without needing to log into the Appwrite console. You can use natural conversation to list entire database schemas, inspect user registration trends, or check system health metrics instantly.

A+ Quality Score 100/100

backend-as-a-service | authentication | database-management | cloud-storage | open-source | api-backend

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/appwrite

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.

Appwrite MCP

9 tools available
Cloud-hosted on Vinkius

Stop switching between tabs and dashboards just to get a project status update. This MCP lets your AI agent act as a dedicated backend developer for your entire infrastructure. Instead of manually checking the console for data structure issues or running separate queries to see who's logged in, you simply ask your agent to look it up. Your agent handles everything: listing all databases and collecting documents from specific collections; retrieving user details to spot unusual registration spikes; or auditing storage by accessing buckets and files.

It’s about getting visibility into the guts of your system through plain language. Whether you're tracking how often a cloud function runs, or just need a real-time health check across all services, this MCP gives you that control. You connect via Vinkius, giving your agent instant access to manage and audit every part of your backend infrastructure.

This means you don’t have to remember complex CLI commands. Your agent handles the complexity so you can focus on what matters: building great products.

Core Capabilities

01 — Audit Database Schemas

List all available databases and retrieve specific document collections for structural analysis.

02 — Monitor User Activity

Retrieve lists of project users to track account statuses, monitor registrations, or audit user roles.

03 — Manage Cloud Storage Assets

List storage buckets and iterate through files inside them for a comprehensive asset inventory.

04 — Track Function Operations

View configured cloud functions and list recent execution logs to debug performance issues.

05 — Assess System Health

Get an immediate, real-time health status report across all connected Appwrite services.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and provide your unique Appwrite Project ID, API Key, and Endpoint credentials.

02 Connect the MCP to your preferred AI client (like Cursor or Claude).

03 Ask your agent a question like, 'What is the current health status of the project?' and let it perform the necessary backend operations.

The bottom line is that you use natural conversation to execute complex infrastructure commands without writing code.

Built For

This MCP saves time for full-stack developers who hate context switching. It's essential for backend engineers needing automated infrastructure checks and operations managers who need deep visibility into cloud storage usage without logging into a console.

Backend Engineer

Uses the MCP to automatically monitor function executions or list databases during development cycles.

Operations Manager

Runs scheduled checks on project health and tracks storage usage across all environments without manual console logins.

Full-stack Developer

Quickly retrieves database schemas or audits user collections to validate data integrity before committing code.

What Changes When You Connect

01 Instant system visibility: Instead of digging through multiple dashboards, you can use the `get_health_status` tool to get a single, real-time report on your project's health.

02 Deep data auditing: You can list all databases using `list_databases` and then inspect collections via `list_collections`, giving you immediate structural insights.

03 Automated user tracking: The `list_users` tool lets you monitor registration trends or check account statuses without running manual reports in the console.

04 Storage management confidence: By listing storage buckets (`list_storage_buckets`) and then files inside them (`list_bucket_files`), you maintain a clear, auditable asset inventory.

05 Debugging efficiency: You can run `list_function_executions` to quickly check recent logs for any cloud function, cutting debugging time from hours to minutes.

Real-World Applications

Investigating Data Discrepancies

A data scientist suspects a collection is missing records. They ask their agent to run `list_databases`, find the correct database, and then use `list_collections` followed by `list_documents` to audit the structure and verify the required data points.

Onboarding a New Team Member

An operations manager needs to know what services are running. They ask their agent to use `list_functions` and `get_health_status`. The agent responds by showing the entire service inventory and confirming that all components are online.

Securing User Accounts

A security auditor needs to check for inactive or suspicious accounts. They ask their agent to run `list_users`, which immediately returns a list of users they can then analyze for potential misuse.

Auditing Media Assets

The product owner wants to know the total size and count of stored media. The agent uses `list_storage_buckets` first, followed by `list_bucket_files`, providing a full inventory of all assets.

Patterns to Avoid

Trying to get one status check

✗ AVOID

A user tries to ask the agent for 'the project details' and gets vague responses that don't specify if they mean health, users, or storage.

✓ INSTEAD

Be specific. Ask the agent to run a single tool, like `get_health_status`, or group related tools: 'First, list all databases; then, check the status of those services.'

Forgetting required credentials

✗ AVOID

The user fails to provide the Appwrite Project ID and API Key, causing the agent to fail with a generic authentication error.

✓ INSTEAD

Ensure you connect this MCP using all three required credentials (Project ID, API Key, Endpoint) before asking for any data retrieval.

Mixing up function logs

✗ AVOID

The user asks about 'last runs' but doesn't specify if they mean the general project health or a specific cloud function's execution history.

✓ INSTEAD

Use `list_function_executions` when you need to track runtime failures, and use `get_health_status` for overall service availability.

The Right Fit

Use this MCP if your workflow requires deep, multi-faceted checks into a backend system. You should connect it when you need to run audits—checking user counts via `list_users`, verifying file existence using `list_bucket_files`, or checking infrastructure status with `get_health_status`. Don't use it if you only need simple CRUD operations; for that, stick to direct API calls. Also, don't use this if your primary goal is simply writing code; use a dedicated coding assistant instead. This MCP shines when the job requires connecting multiple data points—for example, listing databases and then checking user permissions across them.

Appwrite MCP for AI Agents: Database Schema Auditing

Today, auditing a backend database is tedious. You have to log into the Appwrite console, navigate through different projects, run schema comparison tools, and manually collect lists of every collection and document type just to confirm data integrity.

With this MCP, you simply ask your agent to audit the structure. It uses `list_databases` to scope out all available databases, then runs `list_collections` to drill into specific schemas, gathering all the necessary metadata in one go.

Appwrite MCP for AI Agents: Monitoring User and Storage Resources

Manually monitoring user growth or storage capacity means constantly cross-referencing multiple dashboards. You check the 'Users' tab, then you switch over to the 'Storage' metrics page, forcing you to copy data between unrelated views.

Now, your agent handles it. It can use `list_users` and `list_storage_buckets` sequentially. This gives you a consolidated view of both account activity and physical assets, all without leaving your chat window.

9 Tools: Database & Storage Operations with Appwrite MCP

Use these tools in natural language prompts to list system resources, audit user data, or retrieve operational logs from your backend.

#	TOOL	DESCRIPTION
01	`get_health_status`	Checks the overall operational health status of your Appwrite project.
02	`list_storage_buckets`	Provides a list of all configured storage buckets in your project.
03	`list_collections`	Lists the specific collections available within a chosen database.
04	`list_databases`	Retrieves an exhaustive list of all databases created in your project.
05	`list_documents`	Lists the documents contained within a specified collection for review.
06	`list_function_executions`	Retrieves recent logs and status reports for executed cloud functions.
07	`list_bucket_files`	Lists the individual files stored inside a specific storage bucket.
08	`list_functions`	Provides an inventory of all defined cloud functions in your system.
09	`list_users`	Retrieves a list of project users, allowing you to inspect account statuses.

See It in Action

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

U What's the current status of the whole project, including users and storage?

Project Health Report

- ✅ **Database Status:** Operational (3/3 databases active)
- 👤 **User Accounts:** 1,245 total registered. Last 7 days saw an increase of 89 accounts.
- 💾 **Storage Usage:** 4.7 GB used across 5 buckets. No buckets are nearing capacity limits.

Overall status: Green. All core services are running normally.

U Can you check the logs for my image resize function?

Retrieving logs for `resize-image` ...

RUN ID	STATUS	DURATION	NOTES
20240612_A	Success	150ms	Image processed correctly.
20240612_B	Failure	20ms	Error: File not found at source path.

I recommend checking the file paths used for this function.

U List all databases and show me the documents in 'product-catalog'

Database Inventory

You have 4 active databases:

1. `production-db` (Primary data)
2. `staging-db` (Testing environment)
3. `analytics-store` (Metrics collection)
4. `user-profiles` (User metadata)

Contents of 'product-catalog' in production-db:

- Document ID: p1001, Title: Widget Deluxe, Status: Active
- Document ID: p2055, Title: Gadget Mini, Status: Discontinued

Frequently Asked Questions

01 **How does the Appwrite MCP help me manage my backend infrastructure?**

The Appwrite MCP gives your AI agent command-line access to all your core services. You can ask it to audit databases, check storage usage, or monitor function logs without ever leaving your chat window. It centralizes visibility into everything you build.

02 **What kind of data can I retrieve using the Appwrite MCP?**

You can get a wide range of structured data: lists of all databases and collections, user registration records, file inventories from storage buckets, and real-time project health metrics. It's comprehensive backend data.

03 **Is the Appwrite MCP suitable for large production systems?**

Yes. Because it connects to your live project credentials, you can use it to run critical audits, like checking function execution logs and monitoring user activity in a high-volume environment.

04 **Does the Appwrite MCP help me find bugs?**

Absolutely. You can ask your agent to list recent cloud function executions. This lets you quickly identify failures, check error messages, and pinpoint which services need attention for debugging.

05 **Do I have to be a developer to use the Appwrite MCP?**

No. You just need to know what data you're looking for. The agent handles the technical jargon; you talk to it like you're talking to a teammate about system status.

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 `"appwrite": { "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

Appwrite 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 Appwrite. 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	Appwrite MCP
Server ID	`019d8417-e3c4-7053-9782-f424304f0b78`
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/appwrite.