MCP SERVER | NO CODE | CLOUD HOSTED

Sanity MCP

Control your entire CMS from chat, not a UI.

Sanity provides full control over your headless CMS content by letting any AI agent perform operations through natural conversation. This MCP lets you query documents using custom GROQ queries, create new drafts, update existing fields, and even manage entire datasets without ever opening the Sanity Studio interface. You can list all project members or browse media assets—it turns complex backend administration into simple chat commands.

A+ Quality Score 100/100

structured-content | groq | content-api | data-modeling | real-time-sync

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

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.

Sanity MCP

11 tools available
Cloud-hosted on Vinkius

Your AI agent now acts as a dedicated content operations assistant for your Sanity project. Instead of navigating multiple dashboards to find a document ID or remembering precise query syntax, you just tell your client what you need done.

It handles everything from drafting new blog posts and updating product details to running complex data checks across different datasets. Need to know which users have access? Or maybe you're staging content and need to create a clean dataset for development testing? Your agent manages it all, keeping track of your entire content lifecycle in one place.

Connecting this MCP through the Vinkius catalog gives any compatible AI client instant, conversational access to your entire content graph. You manage data with plain language commands; the system handles the complex API calls.

Core Capabilities

01 — Run custom content queries

Ask the agent to search and retrieve documents by specific criteria using advanced GROQ query syntax.

02 — Create, update, and delete records

Add new documents or modify existing fields (like changing a status from draft to published) with single commands.

03 — Manage content collections

Browse all available document types within your project scope.

04 — Administer data environments

List existing datasets or create new ones, isolating content for different stages like staging or development.

05 — Handle media assets and users

Browse your image library and audit who has access to the project's backend tools.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP, then provide your Sanity Project ID, Dataset name, and API Token.

02 Connect your preferred AI client (like Cursor or Claude) to the Vinkius catalog. This gives your agent access credentials.

03 Tell your agent what you need—for example, 'List all drafts of product documents'—and it executes the necessary commands.

The bottom line is that your AI client treats complex content management like a simple conversation with an expert assistant.

Built For

This MCP is for technical content teams, developers building data-driven sites, and senior editors who spend too much time jumping between UI screens. It's perfect for anyone who gets frustrated when a simple task requires navigating five different tabs just to run one query.

Content Editor

Searches for specific articles by keyword, reviews the full details of existing documents, and manages media assets without leaving their writing interface.

Developer/Engineer

Runs complex GROQ queries to validate data structures or programmatically creates new document drafts with required schemas for testing purposes.

Content Operations Manager (Ops)

Audits project access by listing all users and manages the creation of isolated datasets, ensuring development content doesn't mess up production data.

What Changes When You Connect

01 You eliminate the need to manually run complex queries. Instead of writing intricate GROQ code every time you need data, simply ask your agent to query documents using `query_documents` and get instant results.

02 Content publishing becomes conversational. You can draft content or change a document's status from 'draft' to 'published' by asking the agent to run an `update_document` command, skipping clicks entirely.

03 Data isolation is effortless. If you need to test changes without affecting live data, use the MCP to list and create dedicated datasets via `list_datasets` and `create_dataset`.

04 Auditing access used to mean clicking through user roles. Now, you simply ask your agent to run `list_users`, getting a clean roster of who has access instantly.

05 Media management is direct. Instead of navigating the asset library manually, you can list images and their metadata by running `list_image_assets` directly from your chat window.

Real-World Applications

Finding all articles needing a content review

A Content Editor needs to find every product post that was written last week but hasn't been assigned an image yet. They ask their agent to run a query, and the MCP executes `query_documents` using filters for 'product' type and date criteria, returning only the IDs of articles needing review.

Setting up a clean environment for QA

A Developer is working on a new feature and needs to test content against production data. They first ask their agent to `list_datasets` to verify 'production' exists, then command it to `create_dataset` named 'qa-test', ensuring zero risk to live content.

Mass updating product details

A Marketing Manager needs to change the status of 15 old blog posts from draft to published. Rather than opening 15 documents, they instruct their agent to `update_document` for a list of IDs and set the 'status' field to 'published'.

Retrieving specific document details by ID

An Editor gets an email with only a content item ID. Instead of guessing where it lives, they ask their agent to `get_document` using the provided ID and instantly retrieve all associated fields for review.

Patterns to Avoid

Trying to manually list documents

✗ AVOID

The user tries to guess if a document exists by running multiple, slightly different search queries across the API.

✓ INSTEAD

Instead of guessing, tell your agent to `list_documents` with the specific content type you are looking for (e.g., 'post'). This gives you a structured list of all existing records.

Overwriting critical fields

✗ AVOID

A developer runs an update command without specifying which field needs changing, potentially wiping out valuable metadata.

✓ INSTEAD

Always use `update_document` and provide a JSON object that specifies only the exact fields you want to modify. The agent ensures no other data is accidentally changed.

Confusing datasets

✗ AVOID

A team member makes changes in the wrong environment, inadvertently publishing test content directly to the live site.

✓ INSTEAD

Before making any write operation, always use `list_datasets` and confirm you are operating against the intended dataset (e.g., 'development') by using `create_dataset` first.

The Right Fit

Use this MCP if your primary bottleneck is *content workflow* or *data manipulation*. Specifically, if you need to perform CRUD operations on headless CMS content—creating drafts, modifying fields, deleting old articles—then this is the tool. It lets your agent act as a full-stack content editor.

Don't use it if all you need is simple data retrieval for display purposes (e.g., 'give me the title and slug of every post'). In that case, a read-only API connector might be simpler and faster to integrate. But because this MCP handles writes (`create_document` , `update_document`) alongside reads (`query_documents`), it's ideal when your task requires making changes based on data discovery.

If your goal is just asset management without content, you probably need a dedicated media library tool instead of one that includes document creation.

The Pain of Context Switching

Today, managing content requires constant context switching. To update a single product page, you might have to jump from your writing app into the CMS UI, find the correct document ID in one tab, run a query in another, and finally navigate back out to save the changes. This clicking dance is slow and error-prone.

With this MCP, that entire process collapses. You just speak to your agent: 'Update product ABC's price to $299.' The agent handles finding the document ID, making sure you're in the right dataset, updating only the price field, and confirming it—all without you ever leaving your chat window.

Content Management with Sanity

The ability to run a complex search is massive. Instead of manually building a filter or running multiple searches for specific document types, the agent executes a precise `query_documents` command based on natural language input.

This means you're not just querying data; you’re getting actionable intelligence instantly. Your AI client has full control over your content lifecycle—it moves beyond simple retrieval to become a true operational partner.

Sanity Alternative: 12 Tools

These tools let you perform every common task in headless CMS management—from drafting content to auditing user access—all through your AI chat interface.

#	TOOL	DESCRIPTION
01	`create_dataset`	Creates an empty, isolated content collection for testing purposes (e.g., 'staging').
02	`create_document`	Generates and saves a brand new document with specified type fields.
03	`delete_document`	Permanently removes an existing content item from the database; this action cannot be undone.
04	`get_document`	Retrieves all field data for a single, specific document using its unique ID.
05	`list_datasets`	Displays all the isolated content environments currently configured in your project.
06	`list_documents`	Lists every available document of a specific type within the current dataset, including metadata.
07	`list_image_assets`	Retrieves metadata for all images and media files stored in your content library.
08	`list_users`	Lists every user account that has access to the project, useful for security audits.
09	`query_documents`	Runs highly customized GROQ queries across your content to find data based on complex logic (e.g., 'all posts published last month').
10	`search_documents`	Performs a broader search of documents, optionally filtering by type and setting a maximum result count.
11	`update_document`	Modifies specific fields (like title or status) on an existing document without affecting other content.

See It in Action

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

U List all published blog posts in my Sanity project.

I found 15 documents of type 'post'. The most recent include: 'Getting Started with Sanity' (published 2 days ago), 'GROQ Query Tips' (published last week), and 'Content Modeling Best Practices' (published 2 weeks ago).

U Create a new draft post titled 'API Integration Guide' with status 'draft'.

Done! I've created a new document of type 'post' with title 'API Integration Guide' and status 'draft'. The document ID is drafts.abc123def456. It's ready for editing in Sanity Studio.

U Run a GROQ query to find all products with price greater than 100.

Found 8 products with price > 100. The most expensive is 'Enterprise Plan' at $499/month, followed by 'Pro Plan' at $199/month. Results include product name, price and slug for each.

Frequently Asked Questions

01 **How do I run custom searches with Sanity using the Sanity MCP?**

You use the `query_documents` tool. You simply ask your agent what data you need, and it translates that into a powerful GROQ query to pull exactly what you're looking for.

02 **Can I create content drafts with Sanity using this MCP?**

Yes, you use the `create_document` tool. You can specify the document type and fill in basic content fields (like a title) to generate a new draft ready for editing.

03 **Is it safe to delete documents using the Sanity MCP?**

The `delete_document` action is irreversible. Always confirm with your agent that you have the correct document ID and are sure you want to permanently remove the content before confirming.

04 **What if I need to test changes in a sandbox environment?**

You should use `list_datasets` first, then run `create_dataset` to make an isolated copy. This ensures any testing or mistakes won't affect your live production content.

05 **How do I check who has access to my Sanity project?**

Simply ask the agent to execute the `list_users` tool. It will retrieve a list of all registered team members and their current permissions within the project.

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 `"sanity-alternative": { "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

Sanity 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 Sanity. 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	Sanity MCP
Server ID	`019d847b-2e55-70da-9820-8e8800be4293`
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/sanity-alternative.