MCP SERVER | NO CODE | CLOUD HOSTED

Novu MCP

Manage multi-channel notifications conversationally.

Novu lets you manage all your multi-channel notifications—email, SMS, push, and chat—directly through conversation. Trigger complex workflows, update subscriber profiles in bulk, and audit notification preferences without ever opening a dashboard or writing boilerplate API calls.

A+ Quality Score 98.33/100

notification-infrastructure | multi-channel | subscriber-management | event-triggering | push-notifications | workflow-automation

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

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.

Novu MCP

39 tools available
Cloud-hosted on Vinkius

You've got critical communication flows running across multiple channels, but managing them usually means jumping between Postman, your database, and the Novu UI. This MCP changes that. It connects your existing AI client to Novu's entire notification infrastructure, letting you treat complex workflows like a conversation.

Need to test a new welcome email campaign? You can now use natural language commands to trigger single events or bulk blasts. Need to fix stale user data? Just ask the agent to search for users and update their preferences in one go. It handles everything from creating entire notification topics to canceling delayed workflows using just conversational prompts.

Connecting this MCP through Vinkius means you don't have to manually write out every API call; your AI client acts as a dedicated communications engineer, giving you full control over subscriber management and event triggering right where you're already working.

Core Capabilities

01 — Triggering multi-channel events

Send notifications to single users or entire groups across email, SMS, push, and chat by naming the workflow.

02 — Managing subscriber profiles

Search for users by email or ID, retrieve their current preferences, and update user details in bulk.

03 — Auditing communication settings

Check if a specific user is subscribed to a topic and view all active integrations attached to the account.

04 — Controlling workflows

Cancel pending or active notification flows, like scheduled digests or delayed messages, using a transaction ID.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your unique Novu Secret Key.

02 Your AI client accesses the service, allowing you to interact with all subscriber and workflow tools conversationally.

03 You issue a natural language command (e.g., 'Send a welcome email to John Doe') and the agent executes the necessary sequence of API calls.

The bottom line is you use your AI client's conversational ability instead of writing complex, multi-step code or hitting multiple endpoints.

Built For

This MCP is for the Product Manager who needs to verify workflow payloads without asking a developer. It's for Growth Teams running campaigns that require rapid A/B testing of message delivery, and for Developers tired of switching between Postman and their IDE just to test an endpoint.

Product Manager

Verifying complex workflow payloads or checking specific subscriber preferences before launching a new feature.

Growth Marketing Specialist

Running rapid, conversational tests on broadcast notifications to gauge campaign impact instantly.

Software Developer

Testing notification triggers and subscriber data updates directly from their code editor without leaving the development environment.

What Changes When You Connect

01 Automate complex campaign testing. Instead of writing separate code to test welcome emails versus SMS blasts, you simply tell your agent which workflows to `trigger_bulk_event` for a group of users.

02 Fix user data issues instantly. When a Product Manager finds an outdated email address, they can use the agent to first `search_subscribers` and then immediately `update_subscriber_preference`, all without leaving their chat window.

03 Eliminate manual API calls. You don't need to manually build payloads or switch between Postman endpoints; your AI client handles the complexity of calling tools like `trigger_event` for you.

04 Maintain compliance easily. Use the MCP to retrieve and inspect every user's settings via `get_subscriber_preferences`, ensuring all communication adheres to their stated preferences before a blast goes out.

05 Full lifecycle control. You can manage everything from creating new notification topics (`create_topic`) to deleting entire outdated integrations (`delete_integration`), giving you full administrative oversight.

Real-World Applications

Onboarding flow verification

A Product Manager needs to confirm that a user gets the correct sequence of notifications (welcome email, followed by a push notification) when they sign up. They ask their agent to `trigger_event` for the 'onboarding' workflow and can immediately verify if all channels fired correctly.

Marketing campaign rollout

A Growth Team needs to send an urgent update about a new feature to 10,000 users. Instead of writing a massive script, they instruct their agent to `trigger_broadcast` and instantly check the success rate via the returned logs.

Data cleanup project

The Ops Engineer realizes several key subscribers have old credentials or stale preferences. They use the MCP to first `search_subscribers`, then call `update_subscriber` and `upsert_subscriber_credentials` for all identified accounts in a single conversational flow.

Testing complex dependencies

A developer needs to ensure that when Topic A is created, the related topic subscription is also handled correctly. They use the MCP to first `create_topic`, and then immediately follow up by calling `create_topic_subscriptions` for a test user.

Patterns to Avoid

Using multiple tabs/tools

✗ AVOID

A user manually copies IDs from the Novu UI, pastes them into Postman to hit an endpoint, then logs back into their IDE to write the code that uses those values. This is slow and error-prone.

✓ INSTEAD

Connect this MCP via Vinkius. Instead of copying IDs, you tell your agent: 'Cancel the notification for transaction ID X.' The agent handles the entire process conversationally, keeping everything in one chat window.

Ignoring user preferences

✗ AVOID

Running a massive `trigger_broadcast` without first checking if users opted out of SMS. This results in immediate compliance risk and bad PR.

✓ INSTEAD

Before triggering, ask your agent to run 'Check the subscriber for topic X' using `check_topic_subscriber`. The AI will validate consent before sending any message.

Over-engineering workflows

✗ AVOID

Writing an overly complex script that handles both creating a user and updating their preferences. If one step fails, the whole thing breaks.

✓ INSTEAD

Use the dedicated tools. First, call `create_subscriber`. Then, in a follow-up command, use `update_subscriber` to handle any necessary profile changes, keeping the actions clean and segmented.

The Right Fit

You should use this MCP if your core problem is managing *delivery* and *user communication*. If you need to send emails, manage opt-ins, or trigger events based on user activity, this is your tool. It

handles the complexity of multi-channel delivery and subscriber data management.

Don't use this MCP if your primary goal is general CRM record keeping, complex financial modeling, or storing unstructured text documents; you need a dedicated database connector for that. Similarly, if you only need to read basic user profiles without worrying about communication preferences, the `get_subscriber` tool is too narrow. Focus on actions related to 'sending' and 'managing consent.'

The manual nightmare of campaign testing.

Today, running a basic A/B test for your marketing team is a multi-step chore. You have to jump into the Novu dashboard to find the user IDs. Then you copy those IDs out and paste them into Postman, manually changing headers or bodies just to simulate sending an SMS vs. an email. If you need to change the payload, it's another round trip, forcing you to switch context constantly.

With this MCP, your agent handles all that mess for you. You simply tell your AI client: 'Test a push notification blast and then follow up with a bulk update of preferences for those users.' The entire sequence runs conversationally, giving you instant results without ever leaving your chat.

Control everything through Novu’s communication tools.

The manual steps that vanish include finding the right tool for every action. You no longer need to manually initiate a `trigger_broadcast` and then remember to check the logs; you can ask the agent to execute both actions in sequence, along with any necessary cleanup like calling `delete_message` afterwards.

What's different now is that your communication infrastructure moves from being a series of disconnected dashboards and APIs into one single point of control—your AI chat. You manage it conversationally.

Novu: 39 Tools for Notification Infrastructure

These tools give you direct, programmatic access to every aspect of your notification flow, from creating new topics to triggering global broadcasts.

#	TOOL	DESCRIPTION
01	bulk_update_subscriber_preferences	Updates communication settings for many subscribers at once.
02	cancel_trigger	Stops active or scheduled notification workflows like digests.
03	check_topic_subscriber	Checks if a specific user has opted in to receive updates about a topic.
04	create_environment_variable	Sets up environment variables needed for the system to function.
05	create_integration	Adds a new connection point, like connecting Slack or Stripe, to Novu.
06	create_layout	Builds the visual structure or template for notifications.
07	create_subscriber	Registers a brand new user into the system.
08	create_topic_subscriptions	Signs up an existing subscriber to receive messages about a specific topic.
09	create_topic	Creates a new subject or category for notifications.
10	create_workflow	Builds a completely new, complex notification sequence (a workflow).
11	delete_integration	Removes an existing connection point from the account.
12	delete_message	Deletes a specific message record that was sent out.
13	delete_messages_by_transaction	Removes multiple messages using a unique transaction ID.
14	delete_subscriber_credentials	Wipes the login credentials for a subscriber.
15	delete_subscriber	Permanently removes a user from the platform.
16	delete_topic_subscriptions	Unsubscribes a user from all messages related to a specific topic.

#	TOOL	DESCRIPTION
17	`delete_topic`	Deletes an entire notification topic and its history.
18	`get_environment_variable_usage`	Shows how often a specific environment variable is being used by the system.
19	`get_subscriber_preferences`	Retrieves all communication settings and preferences for one user.
20	`get_subscriber`	Finds a subscriber's profile using their unique ID.
21	`get_topic_subscription`	Retrieves the details of one user's subscription to a topic.
22	`get_topic`	Looks up information about a specific notification topic by its key.
23	`list_active_integrations`	Lists all connections (like Slack or Salesforce) currently active with Novu.
24	`list_environment_variables`	Retrieves a list of all configured environment variables.
25	`list_integrations`	Shows every integration that can be connected to Novu.
26	`list_messages`	Retrieves a list of messages sent through the system.
27	`list_topic_subscriptions`	Lists all users currently subscribed to updates for one topic.
28	`search_subscribers`	Finds subscribers using search criteria like email address or name.
29	`set_primary_integration`	Designates a specific connection (e.g., Email) as the main communication channel.
30	`trigger_broadcast`	Sends an immediate notification to every single subscriber registered in the system.
31	`trigger_bulk_event`	Triggers multiple separate notifications and events in a single request.
32	`trigger_event`	Sends a specific notification to one or more targeted subscribers.
33	`update_environment_variable`	Changes the value of an existing environment variable.
34	`update_integration`	Modifies settings for a connected third-party service.
35	`update_subscriber_credentials`	Updates the login credentials stored for a subscriber.

#	TOOL	DESCRIPTION
36	update_subscriber_preference	Changes one specific communication setting, like changing SMS opt-in status.
37	update_subscriber	Updates core profile information for an existing subscriber.
38	update_topic	Modifies the name or description of a notification topic.
39	upsert_subscriber_credentials	Creates or updates credentials for a subscriber in one step.

See It in Action

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

U Trigger the 'welcome-onboarding' workflow for subscriber 'user_99' with the payload { "name": "Alice" }.

I've triggered the 'welcome-onboarding' event for subscriber 'user_99'. The transaction has been initiated successfully.

U Search for a subscriber with the email 'support@example.com'.

I found a subscriber matching that email. The ID is 'sub_5544' and the name is 'Support Team'. Would you like to see their full preferences?

U Cancel the notification trigger with transaction ID 'tx_abc123'.

The workflow execution for transaction 'tx_abc123' has been successfully cancelled.

Frequently Asked Questions

01 **How do I trigger an event using the Novu MCP?**

You use the `trigger_event` tool by describing the workflow name and providing any required payload data. This sends a targeted notification to specific subscribers or groups.

02 **Can Novu MCP handle bulk updates?**

Yes, you can perform bulk actions like updating preferences for many users at once using `bulk_update_subscriber_preferences`, saving time compared to individual API calls.

03 **Is the Novu MCP only for email notifications?**

No. This MCP handles multi-channel delivery, allowing you to manage SMS, push, and chat notifications as well as traditional emails.

04 **How do I find a specific user's details using the Novu MCP?**

You can use `search_subscribers` if you have an email or partial name, or use `get_subscriber` if you already know their unique ID.

05 **What if I need to stop a pending notification? Does Novu MCP support that?**

Yes. You can cancel active or delayed flows using the `cancel_trigger` tool, which requires a specific transaction ID for accuracy.

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 `"novu": { "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

Novu 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 Novu. 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	Novu MCP
Server ID	`019e38c9-9e2a-71b3-ad54-55cc6949a123`
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/novu.