

MCP SERVER

NO CODE

CLOUD HOSTED

Discord Bot MCP for AI Agents

Manage Guilds and Channels from a Natural Conversation

The Discord Bot MCP gives your AI agent total control over a community's backend systems. You can audit entire guilds, review channel structures, manage member lists, and send or delete messages directly from conversation. Stop jumping between dashboards; let your agent handle all real-time communication tasks.

A+ Quality Score 98.33/100

bot-management

server-administration

community-engagement

webhook

real-time-messaging



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

Discord Bot MCP

10 tools available

Cloud-hosted on Vinkius

Managing a large Discord community used to mean opening multiple developer consoles, manually running API calls, or wading through complex web dashboards. Now, you can just ask your AI client to do it. This MCP connects your bot's full power directly into your conversational workflow.

Your agent acts as an instant moderator and administrator. Need to see every channel in a guild? Just ask. Want to know who is in the server or if someone was banned? Your agent checks the member lists and tells you. You can also send announcements across specific channels, or even clean up chat by deleting old messages.

Whether you're running a support team or developing bot features, this MCP lets your agent handle data querying—from listing all guilds your bot belongs to, to getting detailed information about roles. It's basically giving the AI instant admin privileges over your entire community ecosystem. Since Vinkius hosts thousands of specialized MCPs, adding this Discord Bot integration means you have a single point of control for everything related to real-time chat platforms.

Core Capabilities

01 — Audit and list all associated guilds

Retrieves comprehensive metadata for every server the bot is part of, including member counts and role structures.

03 — Manage messages in real time

Sends new announcements or deletes unwanted historical messages in any specified channel.

05 — Retrieve bot and user governance data

Gathers specific information about the bot's own account or other users within the Discord environment.

02 — View and organize channel details

Lists channels within a guild to keep track of the overall community structure.

04 — Inspect member profiles and lists

Pulls up the current list of members for a guild, allowing you to track growth and participant details.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/mcp/discord-bot — connect your AI agent in three steps.

- 01 Subscribe to this MCP on Vinkius and input your unique Discord Bot Token.
- 02 Connect your preferred AI client (Cursor, Claude, etc.) to the configured MCP.
- 03 Ask your agent a question—like 'List all channels in Dev Lounge'—and it executes the command instantly.

The bottom line is, you talk to your AI assistant like talking to a colleague, and it performs complex backend operations on Discord without needing any manual setup or terminal commands.

Built For

This MCP is essential for community managers who spend hours clicking through dashboards. It's also perfect for bot developers who need to test message delivery and read guild metadata without writing boilerplate API scripts. If you manage a large user base on Discord, this is where you need to be.

Community Manager

Uses the MCP to quickly audit channel structures or send mass announcements directly from their chat interface.

Bot Developer

Tests message delivery and retrieves detailed guild metadata—like role permissions—to validate bot functionality without manual API calls.

Support Team Lead

Performs rapid audits of member lists to check user status or sends immediate announcements during an outage.

What Changes When You Connect

- 01 Stop wasting time context-switching. You can use the agent to list all guilds or audit channel structures without ever leaving your chat window.

-
- 02 Maintain full administrative oversight with tools like `list_guild_members` and `get_guild_roles`, giving you a real-time view of who's in the server and what permissions they have.

 - 03 Handle real-time communication tasks immediately. Use `send_message` to make announcements or `delete_message` to clean up spam, all via natural language prompts.

 - 04 The bot can act as an auditor using `get_guild` and `list_channels`, providing immediate metadata about the server's health and organization.

 - 05 Gain full visibility into user status. The agent lets you query member lists or even get specific bot info with `get_me`, making governance simple.
-

Real-World Applications

Emergency Announcement

During a service outage, the community manager doesn't have time to log into Discord settings. They simply ask their agent to 'Send maintenance starting in 10 minutes' to the #announcements channel using `send_message`, ensuring everyone gets the immediate notice.

Tracking Community Growth

The operations lead needs a count of active users for reporting. They prompt their agent to 'List all guild members' using `list_guild_members`, getting a clean, queryable roster right in the chat.

Security Audit

A developer needs to check if a new role ('Premium Member') has been assigned correctly. Instead of checking role permissions manually, they ask their agent to use `get_guild_roles` and verify the permission bits instantly.

Structural Review

A moderator wants to see if new channels were created but forgotten. They ask their agent to list all available channels and get detailed info on a specific one using `list_channels` and `get_channel`, keeping the community organized.

Patterns to Avoid

Using Discord Web Dashboard for Data

✗ AVOID

Manually navigating to the member list, filtering by role, and then copying data into a spreadsheet is slow and prone to missing entries.

✓ INSTEAD

Just ask your agent to 'List all guild members' using ``list_guild_members``. It gathers the full roster instantly and provides structured data right in the chat.

Writing Boilerplate API Calls

✗ AVOID

A developer spends an hour writing a script just to check what roles exist or if a specific message needs deletion.

✓ INSTEAD

Use ``get_guild_roles`` to instantly audit all permissions, or use ``delete_message`` when you spot spam. It's faster than any code.

Guessing Bot Permissions

✗ AVOID

Assuming the bot has access to everything and running commands that fail because it lacks specific scope.

✓ INSTEAD

Use ``get_me`` first to confirm exactly what data your connected bot account is authorized to see. It's good practice.

The Right Fit

You should use this MCP if your workflow requires real-time, conversational interaction with Discord's backend—specifically for auditing channels, managing members, or sending live messages. If you are building a bot that needs to respond dynamically based on server state (like checking role permissions before granting access), this is your tool.

Don't use it if your core need is purely static data storage, like integrating with a database or CRM; for those, look at general record management MCPs. Also, don't rely on this solely for complex moderation logic that requires decision-making beyond simple reads and writes; you might need an agent capable of multi-step reasoning.

If your goal is to track who can do what in the server, use `get_guild_roles` . If you just need a current list of users, stick with `list_guild_members` .

Discord Bot MCP: Streamlining Community Audits and Member Management

Today, checking on your community's health is manual work. You have to jump into Discord, navigate through the server settings, use different panels to list members, then switch tabs to check channel names. It's a painful cycle of clicking, opening new windows, and copy-pasting data just to get a single overview.

With this MCP, you simply ask your agent for an audit. You can request detailed member lists or run through all the channels in a guild using `list_channels` . The AI gathers that metadata instantly and presents it back to you as clean, structured text, saving you twenty minutes of tedious dashboard clicking.

Discord Bot MCP: Controlling Live Messaging and Announcements

When an urgent announcement is needed, or if spam pops up, the manual process is slow. You might have to run a command line tool just to send a message, or spend time manually searching for specific messages that need deleting.

Now you can use your agent like an instant comms officer. Need to delete old chat history? Use `delete_message` . Making a server-wide update? Just ask it to 'Send the announcement' using `send_message` . The outcome is immediate, precise control over your real-time conversation flow.

Discord Bot: 10 Tools for Guild Channel Management

Use these tools with your agent to execute commands that list members, audit roles, delete messages, or send announcements across any Discord server.

#	TOOL	DESCRIPTION
01	<code>list_messages</code>	Fetches and displays the most recent messages sent in a specific channel.
02	<code>delete_message</code>	Removes specified messages from any given Discord channel.
03	<code>get_channel</code>	Provides full details and metadata for a single, specific Discord channel.
04	<code>get_guild_roles</code>	Retrieves a list of all defined roles within a guild, including their associated permissions.
05	<code>get_guild</code>	Gathers comprehensive details about an entire Discord server (guild).
06	<code>get_me</code>	Returns authenticated information and status details for the connected bot account.
07	<code>list_channels</code>	Generates a list of all channels present within a specified guild.
08	<code>list_guild_members</code>	Pulls up the complete member roster for an entire guild, detailing profiles and status.
09	<code>list_guilds</code>	Lists every server that the connected bot account is currently a part of.
10	<code>send_message</code>	Sends custom, live messages into any target channel within Discord.

See It in Action

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

U What are all the guilds my bot belongs to right now?



I checked the system and found that your bot is active in 4 servers. Here's a quick list:

- **Vinkius Community:** (12,500 members)
- **Dev Lounge:** (890 members)
- **Support Hub:** (3,100 members)
- **Marketing Team:** (450 members)

You can ask me to audit any of these next.

U Send a reminder about the meeting tomorrow in the #general channel.



REMINDER: The cross-functional team sync is scheduled for 9 AM EST tomorrow. Don't forget your prep work!

(Message sent successfully to #general. ID: 789012345)

U Who are the top 5 administrators in the main server?



I pulled the member list and checked roles for you. The top admins right now are:

- @AdminUser (Role: Owner)
- @SuperMod (Role: Moderator)
- @TeamLeadX (Role: Admin)
- @SupportGuy (Role: VIP)
- @DevMaster (Role: Developer)

Need me to check the specific permissions for any of these roles?

Frequently Asked Questions

01 How do I use the Discord Bot MCP for AI Agents to audit my community?

This MCP lets your agent run full audits on your server. You can ask it to list all guilds or get detailed metadata about channels and roles, giving you a complete picture of your community's structure.

02 Can the Discord Bot MCP help me with daily member management tasks?

Yes. Your agent handles everything from listing every guild member to getting detailed bot user info. It makes tracking growth and managing profiles as simple as asking a question in chat.

03 Is the Discord Bot MCP good for sending out mass announcements?

Absolutely. You can use your agent to send messages to specific channels, perfect for time-sensitive updates or announcements across multiple groups without manual intervention.

04 Does this MCP let me clean up old messages in Discord?

Yes, you can instruct the agent to delete messages from a channel. This is useful for removing spam or archiving outdated chat history when needed.

05 What if I want to check out all the different channels in my server?

You simply tell your agent to list all available channels. It provides an organized catalog of every channel, helping you maintain a clear view of your community's organization.

06 Does connecting the Discord Bot MCP require developer skills?







No. You don't need to write code or use APIs. The AI agent handles all the complex connections; you just talk to it naturally and tell it what action you want done.

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 <code>"discord-bot": { "url": "..."</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Discord Bot 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](https://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 Discord Bot. 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	Discord Bot MCP
Server ID	019d842e-d04e-7181-94be-eabffaafef1c
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/discord-bot.