

MCP SERVER

NO CODE

CLOUD HOSTED

Slack MCP for AI Agents

Manage team communications and workspace history without switching tabs

Slack MCP lets your AI agent join your team conversations. It can send messages, search through old threads, list your channels, and manage workspace members. It's about moving from copy-pasting from Slack to having your agent handle the communication directly.

A+ Quality Score 100/100

messaging

team-chat

workflow-automation

channel-management

notification-management



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 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

Slack MCP

13 tools available

Cloud-hosted on Vinkius

You spend half your day jumping between your workspace and your AI tool. You see a notification, copy a summary from a channel, paste it into a prompt, and then have to manually send a follow-up message to a different team. This Slack MCP changes that. Instead of being a bridge between two apps, your AI client becomes a member of the team. It can read through history to find that one specific decision made three weeks ago, post deployment updates to the right engineering channel, or even react to a teammate's message to acknowledge it. It's about making your AI actually useful in the place where the work happens. When you find this in the Vinkius catalog, you aren't just adding a plugin; you're giving your agent a seat at the table. It handles the heavy lifting of message retrieval and broadcasting so you can stay focused on the actual output. You can ask your agent to summarize a long thread, find a specific user's status, or broadcast an update to everyone at once. It removes the friction of tab-switching and lets your agent do the tedious work of navigating your workspace history.

Core Capabilities

01 — Send messages to channels or DMs

Your agent can post text or threaded replies directly to any Slack destination.

02 — Search workspace history

Find specific messages or keywords across your entire workspace history.

03 — List and browse channels

Get a full list of public and private channels with their topics and members.

04 — Read channel history

Retrieve the most recent messages from a specific channel to catch up quickly.

05 — Manage user profiles

List workspace members and get detailed profile info like emails and timezones.

06 — Add emoji reactions

Have your agent acknowledge messages with reactions without manual input.

07 — Pin and update topics

Keep important messages pinned or update channel topics to keep info current.

One Click on Vinkius — From Prompt to Execution

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

- 01
- Subscribe to the Slack MCP on Vinkius.
- 02
- Add your Slack Bot Token from your app settings to your client.
- 03
- Ask your AI client to search a channel or send a message.

The bottom line is your AI agent gets a direct seat in your Slack workspace.

Built For

This is for the team lead who needs to broadcast updates across ten channels at once, the developer searching for a bug report in a mountain of chat logs, and the ops manager who needs to audit communication patterns without manual clicking.

Team Lead	Software Engineer	Operations Manager
Sends out company-wide broadcasts and summarizes long project threads for the team.	Searches for specific error logs in engineering channels and posts deployment statuses.	Audits channel activity and manages workspace member lists to keep things organized.

What Changes When You Connect

- 01
- Stop manual copy-pasting by letting your AI client read channel history and post updates directly using `messages_send`.
- 02
- Find old decisions faster by using `messages_search` to query your entire workspace history for specific keywords.
- 03
- Keep your team informed with `messages_send` to broadcast updates to multiple channels at once from your AI client.

-
- 04 Organize your workspace better by using `set_channel_topic` and `channel_info` to keep your channel purposes clear.

 - 05 Acknowledge teammates instantly with `reactions_add` so you don't have to leave your current task to send an emoji.

 - 06 Audit your workspace activity by using `users_list` and `channel_history` to see who is active and what is being said.
-

Real-World Applications

Summarizing long project threads

A project manager asks their agent to read `thread_replies` from the `#project-alpha` channel and summarize the three main blockers found in the last 24 hours.

Hunting for old bug reports

An on-call engineer asks the agent to use `messages_search` to find every mention of 'database timeout' from the last week to identify a recurring pattern.

Broadcasting deployment status

An engineer tells their agent to use `messages_send` to post a 'Success' notification to both the `#dev-ops` and `#general` channels after a production push.

Keeping project plans visible

A team lead asks the agent to use `pin_message` on the latest project roadmap in the `#planning` channel so it stays at the top for everyone.

Patterns to Avoid

Trying to edit messages the bot didn't send

✗ AVOID

Asking the agent to 'fix a typo' on a message sent by a human teammate.

✓ INSTEAD

Use `message_update` only on messages that your bot actually sent. For other messages, you must use the standard Slack UI.

Searching without finding a channel ID first

X AVOID

Asking the agent to 'read history' without specifying which channel.

✓ INSTEAD

First ask your agent to use `channels_list` to find the correct channel ID, then provide that ID for the `channels_history` tool.

Accidental history deletion

X AVOID

Asking the agent to 'clear the channel' which might trigger `message_delete` on many messages.

✓ INSTEAD

Only use `message_delete` for specific, individual messages you want removed. This tool does not have a 'clear all' function.

The Right Fit

Use this if you want your AI agent to actually live inside your Slack workspace. It is the best choice when you need your agent to perform frequent, repetitive tasks like searching for old decisions, summarizing long threads, or posting routine updates to multiple channels. It turns your AI into a team member that can see and speak in the same rooms as you. Don't use this if you need to build a complex, multi-step state machine that triggers on every single webhook or requires heavy backend logic. If you need to build a full-scale automated bot with complex 'if-this-then-that' rules for every single interaction, a dedicated integration platform like Zapier or a custom app might be better. This MCP is designed for natural conversation and context retrieval through your AI client. Use `messages_search` to find the info you need and `messages_send` to push it out to the team.

Slack MCP for Team Communication Automation

Most people handle Slack by constantly switching tabs. You see a notification, jump over to Slack, copy a bunch of text, jump back to your AI, and then have to manually format a reply or a summary. It is a fragmented way to work that breaks your focus and wastes time.

This MCP lets your agent do that heavy lifting for you. You just tell it what you need, and it goes into the workspace, grabs the info, and gives you the answer. You stay in your flow while your agent handles the communication.

Slack MCP for Workspace History Management

You can stop manually auditing channel history or hunting for old deployment notes. The agent can scan your entire workspace history and pull out exactly what matters using search modifiers.

It turns Slack from a place you have to check into a data source your agent already knows. You get the answers you need without the manual digging.

Slack MCP: 13 Tools for Team Chat

Give your agent the ability to send messages, search history, and manage channels directly within your Slack workspace.

#	TOOL	DESCRIPTION
01	<code>channels_list</code>	See all public and private channels your bot can access. This helps you find the correct IDs for other actions.
02	<code>channels_history</code>	Get the latest messages from a specific channel in reverse order. Use this to catch up on recent talk.
03	<code>channel_info</code>	Pull details like topics and descriptions for a single channel. This helps you understand the channel's purpose.
04	<code>messages_send</code>	Post a new message or a threaded reply to any channel or DM. You can use rich formatting for your posts.
05	<code>messages_search</code>	Look for specific keywords, users, or channels across the whole workspace. Results come back sorted by recency.
06	<code>message_update</code>	Edit a message that your bot previously sent to a channel. Use this to correct typos or update info.
07	<code>message_delete</code>	Permanently remove a message the bot sent or one in an admin channel. This action is irreversible.
08	<code>users_list</code>	Get a list of everyone in the workspace with their names and statuses. This is useful for finding specific people.
09	<code>user_info</code>	Pull a full profile including email, timezone, and admin status for a user. It requires a specific user ID.
10	<code>reactions_add</code>	Put an emoji reaction on a message using its specific timestamp. This lets your agent acknowledge messages.
11	<code>thread_replies</code>	Pull all replies from a specific message thread in chronological order. This helps you see the full conversation.
12	<code>pin_message</code>	Keep an important message visible in a channel's details panel. This ensures your team sees critical info.
13	<code>set_channel_topic</code>	Change or update the text that appears at the top of a channel. Use this to keep channel descriptions current.

See It in Action

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

U What was the final decision on the API migration?



API Migration Decision

Based on the conversation in **#engineering-decisions**, the team agreed to:

- **Migration Date:** Next Tuesday at 10:00 PM UTC
- Method:** *Blue-green deployment* **Owner:** @sarah_dev

I found this in a thread started yesterday. Would you like me to post a reminder to the channel?

U Notify the #dev-ops channel that the staging site is down.



Done! I've sent the following message to **#dev-ops**:

> **⚠️ Alert:** The staging site is currently experiencing downtime. The team is investigating.

Would you like me to also react with a **🔥** emoji?

U Who is currently in the #marketing channel?



There are currently **14 members** in the **#marketing** channel. Here are the primary leads:

NAME	ROLE
Alice Chen	Creative Director
Bob Smith	Content Lead
Clara Wu	Social Media Manager

I can get the full list of all members if you'd like.

Frequently Asked Questions

01 Can the Slack MCP send messages for me?

Yes, it can post messages to any channel or direct message. Your agent can even send threaded replies to keep conversations organized.

02 Can I search my old Slack history using the Slack MCP?

Yes, your agent can search across your entire workspace for specific keywords, usernames, or channel names to find old decisions or files.

03 Does the Slack MCP work with Claude or Cursor?

Yes, it works with any MCP-compatible client, including Claude, Cursor, Windsurf, and VS Code.

04 How do I let my AI agent see my Slack channels?

Once you subscribe and add your Slack Bot Token, your agent can automatically list and browse all the channels your bot has access to.

05 Can the Slack MCP add emoji reactions?

Yes, your agent can add reactions to specific messages. This is great for acknowledging tasks or providing quick feedback without typing.

06 Can I update channel topics with the Slack MCP?

Yes, your agent can update or set the topic of any channel. This helps keep your workspace organized as projects evolve.

07 Is my Slack data safe with this MCP?

Your data remains secure because you provide your own Slack Bot Token. The agent only accesses the information you authorize through your workspace permissions.

08 How do I get started with Slack MCP?

Subscribe, then create a Slack App at **`api.slack.com/apps → Create New App`**. Go to **`OAuth & Permissions`**, add the required bot scopes (channels:read, chat:write, search:read, users:read), install the app to your workspace, and paste the **`Bot User OAuth Token`** (starts with xoxb-) here. No code, no webhooks — just connect and go.

09 Can my AI agent send messages and reply in threads?

Yes. Your agent can post to any channel the bot has been invited to — including threaded replies. Just say 'post in #engineering that the deploy is complete' or 'reply in the thread about the bug fix'. It supports Slack's full mrkdwn formatting including bold, links, code blocks, and mentions.

10 How can I find a specific conversation from last week?

Tell your agent to search for it — 'find messages about the Q3 budget review' or 'search for messages from @sarah about the client proposal'. The agent uses Slack's search API with support for modifiers like from:user, in:channel, and date ranges. No more scrolling through channels — your agent retrieves the exact messages you need.

11 Can I manage multiple channels and monitor team activity?

Absolutely. List all channels to get an overview of your workspace structure, read history from specific channels to audit conversations, and check team member profiles, statuses, and timezones. Perfect for distributed teams, engineering organizations, and operations managers who need a bird's-eye view of communication across dozens of channels.

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>"slack": { "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

Slack 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 Slack. 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	July 2026
MCP Server	Slack MCP
Server ID	019d7609-2d59-73c2-a477-b02fcff839d7
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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/slack.