MCP SERVER | NO CODE | CLOUD HOSTED

Teams Webhook Notifier MCP

Send structured alerts directly into your chat.

Microsoft Teams Webhook Notifier MCP sends structured messages and alerts directly into your designated Microsoft Teams channels. It provides a secure, zero-permission bridge for your AI agent to communicate critical updates—like deployment statuses or engineering reports—without needing complex corporate API access.

D Quality Score 55/100

webhook | notifications | alerts | adaptive-cards | zero-trust | real-time-messaging

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/microsoft-teams-webhook-notifier

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Microsoft Teams Webhook Notifier MCP

1 tools available
Cloud-hosted on Vinkius

This MCP gives your AI client the ability to drop important notifications straight into specific Teams channels. Think of it as giving your automation process a megaphone that only reaches one room.

Because this tool uses a simple Incoming Webhook URL, you skip the headache of managing enterprise-level permissions or rotating complex API tokens. Your agent doesn't need read access—it just needs permission to speak. This means you can send rich alerts, like those formatted with Adaptive Cards, complete with actionable buttons and data tables, without compromising your organization’s security posture.

The Vinkius catalog makes this simple connection available across any compatible AI client. You get the ability to reliably alert teams about everything from successful deployments to production bugs using pure, contained messaging. It's one of the safest and most straightforward ways to keep critical information visible where it needs to be.

Core Capabilities

01 — Post Rich Alerts

Send messages that include structured data, buttons, and tables instead of plain text.

02 — Broadcast Status Updates

Notify a specific Teams channel about events like deployments or service status changes.

03 — Trigger Engineering Reports

Automatically post detailed technical reports and incident summaries to the team chat.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/microsoft-teams-webhook-notifier** — connect your AI agent in three steps.

01 First, you set up a single Incoming Webhook URL within your target Teams channel. This URL is how the MCP connects.

02 Next, your AI agent calls this MCP and invokes the `send_teams_message` tool, providing either plain text or detailed JSON for rich cards.

03 Finally, the message bypasses complex corporate permissions and appears in the designated Teams channel instantly.

The bottom line is that you use a simple URL to let your agent speak directly into one specific Teams chat without needing massive security credentials.

Built For

DevOps engineers who hate manually updating status dashboards, incident response teams who need immediate alerts across platforms, and software architects building mission-critical notification pipelines.

DevOps Engineer

Uses this MCP to pipe deployment success/failure statuses directly into the team channel so nobody has to check a dashboard manually.

Site Reliability Engineer (SRE)

Configures automated alerts that instantly post detailed error reports or service degradations using structured MessageCards when thresholds are breached.

Software Architect

Integrates the MCP into agent workflows to manage multi-stage notifications, ensuring complex event data reaches the right team chat safely.

What Changes When You Connect

01 Zero Security Risk: You don't need heavy Graph API tokens or complex permissions. This webhook method keeps the connection surgically contained, only allowing messaging out and nothing else.

02 Rich Content Delivery: Don't settle for plain text updates. Your agent uses `send_teams_message` to generate fully interactive Adaptive Cards with buttons and tables.

03 Deployment Agnostic: Whether you're using your agent in a code IDE like Cursor or an autonomous workflow engine like CrewAI, the message delivery remains simple and reliable.

04 Instant Visibility: Critical alerts hit the team chat immediately. Instead of checking logs or dashboards, everyone sees the status update pop up right where they work.

05 Maximum Containment: Because it's a webhook, your agent cannot read corporate emails or snoop on other channels. It only sends messages to what you define.

Real-World Applications

Deployment Complete Alert

The CI/CD pipeline finishes its build and asks the agent to notify Teams. The agent calls `send_teams_message` with a structured card confirming the version number, commit hash, and providing an immediate link to review the logs.

Incident Bug Report

A monitoring service detects an anomaly. The automation triggers the agent, which formats the full stack trace and impact area into a rich MessageCard using `send_teams_message`, ensuring the on-call team gets all necessary data immediately.

Feature Flag Rollout Status

When a new feature is flipped live, the agent sends a status update to the #releases channel. This message uses `send_teams_message` to include an expiration date and the responsible team contact.

Daily Standup Summary

At 9 AM, the agent aggregates summaries from multiple systems. It then posts a clean, structured daily summary message into the dedicated standup chat using `send_teams_message`.

Patterns to Avoid

Using Full Graph API Integration

✗ AVOID

Attempting to use a broad corporate integration that requires 'read' permissions across all user data or tenants just to send one message.

✓ INSTEAD

Keep it surgical. Use this MCP and the `send_teams_message` tool, which only needs write access to a single, specified webhook URL. It limits blast radius dramatically.

Sending Plain Text Logs

✗ AVOID

Having your agent simply dump raw text logs into Teams that require manual parsing and make no sense in context.

✓ INSTEAD

Always use the optional `cardJson` parameter within `send_teams_message`. This allows you to structure messy data into readable cards with proper headings and tables.

Over-Privileging the Agent

✗ AVOID

Giving your agent general access that lets it read, modify, or delete messages across multiple channels.

✓ INSTEAD

This MCP is built on zero trust. It only provides a webhook endpoint for `send_teams_message`. The agent can't touch anything else in your organization.

The Right Fit

Use this MCP if your primary need is to guarantee that a specific, non-sensitive alert or status update lands reliably and securely into ONE designated Teams channel. If you only need the AI client to broadcast messages (and never read data like user emails or files), this is the perfect fit.

Don't use this if you need your agent to perform actions beyond posting a message, such as reading private chats, fetching details from individual users, or modifying existing records. If those are needs, you must look at other MCPs that offer read tools. This tool only sends; it doesn't know anything else about your corporate environment.

The Problem with Standard Status Updates

Right now, when something happens—say, a critical service goes down—you usually have to jump through five different tabs: the monitoring dashboard, the incident management tool, the Jira ticket board, and then manually copy the key details into Teams. You spend time compiling status updates instead of fixing the problem.

With this MCP, your agent handles that entire communication process for you. You just tell it what happened; the agent uses `send_teams_message` to package all the relevant data—the impact level, the affected service, and the current owner—into one clean, readable card right in Teams.

Send Rich Alerts with send_teams_message

Before this MCP, sending a complex alert meant sticking to basic text formatting, which looks dull and makes it hard for the team to quickly extract what's important. You risked missing critical details because they were buried in paragraphs of raw status text.

Now, your agent uses `send_teams_message` to deliver messages with structured data fields. It means the information is not just written; it's organized into actionable cards that teams can understand at a glance.

Microsoft Teams Webhook Notifier: 1 Tool

You can use the included tool to broadcast simple messages or highly formatted alerts to a specific Microsoft Teams chat.

#	TOOL	DESCRIPTION
01	`send_teams_message`	Sends a notification or message to a Microsoft Teams channel, optionally including rich UI details via JSON.

See It in Action

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

U Notify Teams that the deployment is complete.

I've sent the message 'The deployment is complete.' to the Teams channel.

U Send a rich alert to Teams using a MessageCard format to report a bug.

The rich MessageCard alert detailing the bug has been successfully posted to Teams.

Frequently Asked Questions

01 **Can I use Microsoft Teams Webhook Notifier MCP for general chat?**

No, this MCP only sends messages to the specific channel defined by your incoming webhook URL. It is not designed for general, ad-hoc chatting across multiple locations.

02 **Does send_teams_message require elevated permissions?**

Nope. Because it uses a simple webhook, it bypasses the need for complex corporate API tokens or high-level Graph access, making it very low risk.

03 **What is the limit on what I can send using this MCP?**

You can send plain text messages, but the real power comes from the optional `cardJson` parameter in `send_teams_message`, which lets you create rich Adaptive Cards.

04 **Can my agent read data with Microsoft Teams Webhook Notifier MCP?**

No. This is a one-way street. The MCP only allows your AI client to send messages out; it has zero ability to read or access any other internal corporate data.

05 **Does this work with all types of Teams channels?**

It works for any channel where you can generate a standard Incoming Webhook URL. The setup is specific to the destination chat, not the entire tenant.

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 `"microsoft-teams-webhook-notifier": { "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

Microsoft Teams Webhook Notifier 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 Microsoft Teams Webhook Notifier. 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	Microsoft Teams Webhook Notifier MCP
Server ID	`019e38c1-6bbe-7114-966c-2e4e809938e5`
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/microsoft-teams-webhook-notifier.

© 2026 Vinkius · vinkius.com · support@vinkius.com