MCP SERVER | NO CODE | CLOUD HOSTED

Zoho ZeptoMail MCP

Automate Alerts, Monitor Deliverability.

Zoho ZeptoMail MCP lets your AI agent manage all transactional email needs, from sending password resets to monitoring delivery health. It handles single emails and complex templates using natural conversation, giving you full visibility into who gets your message and whether it actually lands in the inbox.

A+ Quality Score 100/100

transactional-email | otp-delivery | email-templates | delivery-monitoring | smtp-alternative | notification-service

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

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.

Zoho ZeptoMail MCP

7 tools available
Cloud-hosted on Vinkius

Stop opening multiple dashboards just to send an alert or check a bounce rate. This MCP connects directly to Zoho ZeptoMail, letting your AI agent handle all transactional email communications through simple chat commands. You can automate sending emails—whether that's a single notification or a template-based message with merge variables—and manage the entire process without ever logging into the main dashboard. It monitors things you need to know: which domains are active and if any addresses are on a suppression list. Need to check delivery status? Your agent can pull specific logs and full reports, making sure your customer communication is reliable every time. When you connect this MCP through Vinkius, you gain centralized control over the entire lifecycle of an email, letting you focus purely on what message needs to go out next.

Core Capabilities

01 — Send single or templated emails

Draft and dispatch transactional emails using specific HTML/text bodies or pre-approved templates.

02 — Review email delivery status

Check comprehensive logs, including successful deliveries, failures, and detailed reports for any given message.

03 — Manage sending infrastructure

Verify active domains and check suppression lists to ensure your sends won't bounce or get blocked.

04 — Access email blueprints

View and manage existing email templates so you always use consistent branding.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP, then provide your ZeptoMail Send Mail Token in the agent settings.

02 Select the appropriate data center region (like 'US' or 'EU') for your account.

03 Your AI client immediately gains access to all email tools, letting you start automating communications via chat.

The bottom line is that once it's connected, you stop jumping between services and manage all your email comms from one place using natural language prompts.

Built For

This MCP is for developers building notification systems or operations teams tired of manual checks. If your job involves making sure a critical alert, like a password reset or payment confirmation, gets delivered reliably, you need this.

Software Developer

Needs to write automated services for service alerts and user notifications without writing dedicated API wrapper code.

Marketing Operations Specialist

Monitors email template usage and delivery reports to ensure brand consistency and high send rates across campaigns.

Support Team Lead

Quickly checks suppression lists and historical logs to diagnose why a specific customer message failed or bounced.

What Changes When You Connect

01 You can send critical alerts immediately. Use the `send_email` or `send_template_email` tools to dispatch password resets and service notifications directly through conversation, bypassing manual API calls.

02 Delivery tracking is simple. Instead of sifting through raw logs, you can ask your agent to run `list_logs` or `get_log_details` to see exactly when a message was sent and if it succeeded.

03 Maintain sender reputation effortlessly. Run `list_domains` to verify all active sending domains and check the `list_suppression_list` before any major campaign launch.

04 Enforce brand consistency every time. Use `list_templates` first, then select a template for your message; this ensures that every communication uses approved formatting and merge variables.

05 Reduce diagnostic time. Support teams can use these tools to quickly track down delivery failures by checking the logs and identifying if the recipient is on the suppression list.

Real-World Applications

Handling a failed account login notification

A user asks their agent, 'Send a security alert to john@corp.com.' The agent uses `send_email` and confirms the message is queued, guaranteeing immediate communication without developer intervention.

Checking delivery rates after an outage

The support lead asks, 'What were our key deliveries last night?' The agent runs `list_logs`, filtering for errors, and then uses `get_log_details` to confirm the exact reason a specific user couldn't receive their welcome email.

Preparing for a new product launch campaign

The marketing manager first runs `list_templates` to select the correct branding, then uses `list_domains` to verify they have all necessary sending domains active before scheduling mass sends.

Diagnosing recurring bounce addresses

An agent checks if a specific address is blocked by running `list_suppression_list`. If it's listed, the team knows not to bother sending emails until they resolve the underlying issue.

Patterns to Avoid

Using basic email tools for everything

✗ AVOID

A developer might try to send a complex onboarding flow using only a single `send_email` call, resulting in messy code and no template consistency.

✓ INSTEAD

Don't just send raw text. First, use `list_templates` to find the approved structure, then execute `send_template_email`. This keeps your message clean and professional.

Assuming a bounce is user error

✗ AVOID

A support agent manually checks logs but doesn't know if the recipient was preemptively blocked by the system.

✓ INSTEAD

Always check `list_suppression_list` first. If the address shows up there, you know immediately that sending an email won't work and you need to resolve a technical issue before trying again.

Sending without knowing domain status

✗ AVOID

Marketing sends out a major announcement but forgets to verify if their alternate send address is still active.

✓ INSTEAD

Before any large-scale communication, run `list_domains` to confirm all necessary sending domains are verified and ready for use.

The Right Fit

Use this MCP if your primary need is reliable transactional messaging—think alerts, password resets, payment confirmations, or time-sensitive notifications. It excels at automating the *delivery* process itself, allowing you to send messages using consistent templates and verifying sender health along the way. Don't use it if you just want a simple newsletter blast; for bulk marketing campaigns that require deep segmentation or complex list management outside of basic delivery monitoring, an email service provider built purely for mass outreach might be better suited.

However, this is perfect for developers who need to integrate reliable comms into codebases, or support teams who need instant visibility into *why* a message failed. If your workflow involves checking logs (`get_log_details`), managing template content

(`list_templates`), and ensuring the address isn't blocked (`list_suppression_list`), this is exactly what you want.

The headache of manual email checks

Today, checking on your email communication status means jumping between three or four different dashboards. You have to copy a log ID here, navigate to the reports tab there, and cross-reference domains in a third spot just to figure out if 'Client X' got their password reset.

With this MCP, you talk to your agent like talking to a teammate. You ask, 'Did Client X get their email?' Your agent handles all the log pulling, status checking, and domain verification behind the scenes. You get an immediate, accurate answer without touching another dashboard.

Send reliable emails with Zoho ZeptoMail

You don't have to manually check if a message was sent correctly or what happened after it left your system. Your agent manages the complexities of `list_logs` and uses `get_log_details` instantly, giving you full visibility.

It changes everything: Instead of spending 15 minutes debugging delivery failures across multiple systems, you get instant confirmation that the message is sent or exactly why it bounced. It's reliable comms, handled by chat.

Zoho ZeptoMail: 7 Tools for Email Comms

These tools let you manage every aspect of your email comms, from sending a single alert to monitoring complex delivery logs.

#	TOOL	DESCRIPTION
01	`get_log_details`	Pulls complete, detailed information on one specific email log entry.
02	`list_domains`	Lists all the verified sending domains currently active in your account.
03	`list_logs`	Retrieves a list of recent email activity and delivery attempts.
04	`list_suppression_list`	Shows which email addresses are blocked from receiving messages due to bounces or spam reports.
05	`list_templates`	Retrieves a list of all available, pre-designed email templates you can use.
06	`send_email`	Sends a one-off transactional email using plain text or full HTML body content.
07	`send_template_email`	Sends an email by filling out and sending a message based on an existing template.

See It in Action

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

U Send a password reset email to user 'client@example.com' using the 'Password_Reset_V2' template and include their first name in the merge variables.

I successfully sent the password reset email to 'client@example.com' using the 'Password_Reset_V2' template. The merge variables were processed and the email has been queued for delivery.

U Check the email logs to see if the welcome email sent to 'newuser@startup.com' yesterday was successfully delivered.

I checked the delivery logs. The welcome email sent to 'newuser@startup.com' (Log ID: 94829) was successfully delivered yesterday at 14:32 UTC. The recipient also opened the email.

U Verify our active sending domains and check if 'bounced.contact@example.com' is on our suppression list.

Your account has 2 verified sending domains ('api-mailer.net' and 'system.app'). I checked the suppression list and 'bounced.contact@example.com' is currently listed due to a hard bounce last week. You won't be able to send emails to this address.

Frequently Asked Questions

01 **How does Zoho ZeptoMail MCP handle template variables?**

It handles templates using merge variables. When you call `send_template_email`, your agent processes the variable names (like first name or account ID) and fills them into the correct spots in the pre-approved template.

02 **Do I need to worry about spam filters when using send_email?**

The MCP helps by allowing you to monitor your sending reputation. You can use `list_domains` to verify active domains, which is key to maintaining a good sender score.

03 **What if an email address is permanently blocked?**

You check the status using `list_suppression_list`. If the address appears there due to a hard bounce, you know immediately that sending another message will fail and must wait for manual remediation.

04 **Can I see historical data with list_logs?**

Yes, `list_logs` provides a history of all email activity. You can then narrow down the view using `get_log_details` to examine specific messages from past dates.

05 **Does this MCP support different types of emails?**

It supports transactional emails, including single-send notifications via `send_email`, as well as complex, structured communications using the template tools.

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 `"zoho-zeptomail": { "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

Zoho ZeptoMail 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 Zoho ZeptoMail. 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	Zoho ZeptoMail MCP
Server ID	`019dd191-d9d0-701b-ace7-115fd3290680`
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/zoho-zeptomail.