MCP SERVER | NO CODE | CLOUD HOSTED

PagerDuty MCP for AI Agents

Manage production incidents and on-call rotations from your AI client.

PagerDuty MCP. Manage incidents, services, on-call schedules, and escalation policies from any AI agent. Trigger, acknowledge, and resolve alerts without leaving your workspace.

A+ Quality Score 100/100

incident-management | on-call-scheduling | alerting | service-monitoring | escalation-policy | devops

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/pagerduty

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.

PagerDuty MCP

11 tools available
Cloud-hosted on Vinkius

This MCP lets you take control of your PagerDuty incident management through a simple conversation with your AI agent. You can pull up active alerts, see who's currently on-call, or get the full configuration of a monitored service without ever opening a new browser tab. It handles the repetitive tasks of looking up user profiles, listing schedules, and checking escalation policies so you can stay focused on fixing the actual problem. Because it's part of the Vinkius catalog, you can get it running in minutes and start using it across any of your favorite AI clients. It turns your AI into a functional dashboard that responds to your specific commands for triaging and resolving production issues. Instead of hunting through nested menus while a production system is down, you get a direct way to query your status. You can ask your agent to summarize the current situation, find the right person to contact, or update an incident's status with a single command. This takes the friction out of your response workflow and lets you handle high-pressure situations with much less overhead. It's about getting the information you need exactly when you need it, without the constant context switching between your terminal and your management dashboard.

Core Capabilities

01 — Acknowledge and resolve incidents

Update the status of active alerts to move them through your workflow.

02 — List all triggered alerts

Get a full view of every active incident across your services instantly.

03 — Identify who is currently on-call

See exactly who is responsible for active alerts across all schedules.

04 — Browse all monitored services

Inspect the health and configuration of every service in your account.

05 — Inspect escalation policy chains

Understand how alerts route through your teams and who gets notified.

06 — Retrieve specific user profiles

Get contact methods and notification rules for any team member.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to the PagerDuty MCP on Vinkius.

02 Add your PagerDuty REST API Key to the configuration.

03 Ask your AI agent to list incidents or check on-call schedules.

The bottom line is you get a direct line to your incident management platform inside your favorite AI client.

Built For

The SRE who's tired of context-switching between their terminal and the PagerDuty dashboard during a 2 AM outage. It's for people who need to act fast on production issues.

SRE Engineer

Triages active alerts and updates incident statuses while staying in their IDE.

Engineering Manager

Gets a quick pulse on active incidents and on-call coverage without opening a browser.

DevOps Specialist

Automates routine incident updates and checks service health.

What Changes When You Connect

01 Stay in your editor while managing alerts. Use list_incidents to see what needs attention without leaving your workspace.

02 Acknowledge production issues instantly. Use update_incident to claim an alert in seconds through your AI client.

03 Identify the right contact person quickly. Use list_oncalls to see who's active across all your schedules.

04 Check service health on demand. Use get_service to see integration details and status without clicking through menus.

05 Audit your team members easily. Use list_users and get_user to see notification rules and contact info.

06 Map out your alert routing. Use list_escalation_policies to see exactly how incidents move through your teams.

Real-World Applications

Emergency Triage

An SRE asks for all triggered incidents and then acknowledges the most critical one to start the fix.

Shift Handover

A manager asks who is on-call for the Platform team to see who needs to be looped in during a shift change.

Service Auditing

A DevOps lead lists all services to check if a new integration is active and correctly configured.

Incident Resolution

An engineer gets a service's details and then resolves the incident once the fix is live in production.

Patterns to Avoid

Missing required headers

✗ AVOID

Trying to create an incident without a user email will fail.

✓ INSTEAD

Use create_incident with the correct From header email as required by the PagerDuty API.

Asking for general fixes

✗ AVOID

The agent can't fix your code or infrastructure directly.

✓ INSTEAD

Use update_incident to manage the alert's status instead of asking the agent to repair the underlying system.

Guessing service names

✗ AVOID

If the agent can't find a service, it might be using the wrong name.

✓ INSTEAD

Use list_services to get the exact IDs first so your queries are accurate.

The Right Fit

Use this MCP if you need to manage PagerDuty alerts, on-call schedules, or service configurations directly from an AI chat. It's perfect for SREs and DevOps engineers who want to reduce context switching during incidents. Don't use it if you need to perform complex webhooks or heavy API scripting. For those tasks, you're better off using the native PagerDuty API or a dedicated automation script.

PagerDuty MCP for Faster Incident Response

You're in the middle of a production outage. You have to switch from your terminal to a browser, log into PagerDuty, find the right service, identify the incident, and then manually acknowledge it. It's a lot of clicks when every second counts.

With this MCP, you just type Acknowledge the database alert into your AI client. The agent handles the lookup and the status update for you. You stay in your flow while the work gets done.

PagerDuty MCP for Real-Time On-Call Visibility

Managing on-call schedules usually involves digging through nested menus to find out who's on-call for a specific layer or seeing how escalation policies are set up.

This MCP brings that data to your fingertips. You can check schedules and policies instantly, making it much easier to coordinate during high-pressure events.

PagerDuty MCP: 11 Tools for Incident Management

Use these tools to manage alerts, check on-call status, and inspect service configurations through your AI client.

#	TOOL	DESCRIPTION
01	`create_incident`	Create a new incident on a specific service. You just need to provide the service ID, title, and your user email.
02	`get_incident`	Pull up all the details for a specific incident. This is useful for seeing the full history and current status of an alert.
03	`get_service`	View the configuration of a specific service. It shows you the integrations, health status, and other settings.
04	`get_user`	Pull the profile for a specific user. You can see their contact methods and notification rules in one place.
05	`list_oncalls`	See who is currently on-call across all your schedules. This helps you quickly identify the right person to contact.
06	`list_schedules`	List every on-call rotation schedule in your account. You can use this to see coverage windows and handoff times.
07	`list_services`	See a list of every monitored service in your account. It's the fastest way to browse your entire service inventory.
08	`list_users`	Get a list of all users in your PagerDuty account. Use this for auditing your team members or checking permissions.
09	`update_incident`	Change an incident's status to acknowledged, resolved, or escalated. This allows you to manage the lifecycle of an alert programmatically.
10	`list_escalation_policies`	See all the escalation chains for your teams. It helps you understand how incidents route through different groups.
11	`list_incidents`	Pull a list of all incidents across your services. You can even filter them by status like triggered or acknowledged.

See It in Action

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

U Show me all triggered incidents right now.

There are 3 triggered incidents:

ID	STATUS	SERVICE	TITLE
P8K2LMN	🔴 Triggered	Production API	Database connection pool exhausted
P3J9FGH	🔴 Triggered	CDN Edge	SSL certificate expires in 24h
P1A4BCD	🟡 Warning	Background Jobs	High memory usage on worker-03

Would you like me to acknowledge any of these?

U Who is on-call for the Platform team right now?

Current on-call for Platform:

- **Level 1**: sarah.chen@company.com (until Apr 7, 9:00 AM UTC)
- **Level 2**: marcus.ops@company.com (until Apr 14, 9:00 AM UTC)

Escalation Policy: Platform Critical (30 min timeout per level).

U Acknowledge incident P8K2LMN and show me the service details.

✅ Incident **P8K2LMN** acknowledged successfully.

Service: Production API
- **Status**: Active (Critical)
- **Escalation**: Platform Critical (3 levels)
- **Integrations**: Datadog, Sentry, Slack
- **Auto-resolve**: 4 hours

Would you like me to investigate the root cause or resolve this incident?

Frequently Asked Questions

01 **Can I use PagerDuty MCP to acknowledge alerts?**

Yes, you can acknowledge, resolve, or escalate incidents directly through your AI client. This helps you manage the alert lifecycle without leaving your current workspace.

02 **Does PagerDuty MCP show who is on-call?**

Yes, you can see who is currently on-call across all your schedules. It's a fast way to find the right person to contact during an outage.

03 **How do I connect PagerDuty to my AI agent?**

You just need to subscribe to this MCP on Vinkius and add your PagerDuty REST API Key from your account settings. Once connected, your agent can access your data.

04 **Can I see my escalation policies?**

Yes, you can list all escalation policies to see how incidents route through your teams. This gives you a clear picture of your alert routing.

05 **Is PagerDuty MCP good for SRE teams?**

Yes, it's designed for SREs and DevOps teams who need to triage and resolve incidents quickly. It helps reduce the time spent on manual dashboard navigation.

06 **Can I create new incidents with this?**

Yes, you can create new incidents on specific services by providing the title and your user email. This makes it easy to log new issues as they arise.

07 **Can I see my service configurations?**

Yes, you can view the detailed configuration and health status of any monitored service. This is helpful for checking integrations and other settings.

08 **Can I acknowledge and resolve incidents directly from my AI agent?**

Yes! Use the `update_incident` tool with the incident ID and your PagerDuty email. Set status to `acknowledged` or `resolved` to change the incident state instantly without opening the PagerDuty dashboard.

09 **How do I find out who is on-call right now?**

Run the `list_oncalls` tool. It returns every user currently on-call across all schedules and escalation levels, showing their name, escalation policy, and coverage window.

10 **Can I create incidents programmatically for testing or escalation?**

Absolutely. Use `create_incident` with your PagerDuty email, the target service ID, a descriptive title, and optionally set urgency to `high` or `low`. The incident will immediately trigger according to the service's escalation policy.

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 `"pagerduty": { "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

PagerDuty 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 PagerDuty. 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	PagerDuty MCP
Server ID	`019d75ee-2ade-73c6-b403-b225ff91edf1`
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/pagerduty.