MCP SERVER | NO CODE | CLOUD HOSTED

Portainer MCP

Run full DevOps operations via chat.

Portainer MCP connects your AI agent to your entire container infrastructure, letting you manage Docker and Kubernetes environments through natural conversation. You can list containers across multiple endpoints, spin up new services from images, start dormant apps, and connect to remote clusters—all without leaving your chat interface.

A+ Quality Score 100/100

docker | kubernetes | container-management | orchestration | infrastructure-as-code

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/portainer

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.

Portainer MCP

6 tools available
Cloud-hosted on Vinkius

Managing complex container setups usually means jumping between a dashboard, writing `docker-compose` files, or remembering specific CLI commands for different environments. This MCP changes that. It lets you talk to your infrastructure instead of clicking through menus. You can connect your AI agent to any environment—local Docker hosts or remote Kubernetes clusters—and treat them all as one system. Need to check the status of a service on an endpoint you haven't touched in weeks? Just ask. Want to deploy a new testing stack using a specific image? Tell your agent and watch it handle the creation process. Because Vinkius manages this catalog, you connect once from any compatible client and get immediate access to controlling all your core container operations. It’s about giving your AI agent complete operational control over your entire service lifecycle.

Core Capabilities

01 — Check Container Status

List every Docker container running or stopped within a specified environment.

02 — Initialize Environments

Connect your AI agent to new or existing remote and local Docker/Kubernetes clusters (endpoints).

03 — Deploy New Containers

Create a brand-new container instance using a specific image name.

04 — Restart Services

Bring a stopped or dormant container back online with one command.

05 — Secure Access Credentials

Authenticate the MCP connection to generate temporary security tokens for authorized access.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and provide your Portainer URL along with the necessary API key.

02 Next, you instruct your AI agent on what needs doing—for example, 'list all containers in my staging environment.'

03 The system executes the required action against your infrastructure and reports the status directly back to your chat.

The bottom line is that you control complex container operations conversationally instead of through a dashboard or terminal.

Built For

This MCP is for the DevOps Engineer who hates context-switching between dashboards and terminals. It's for developers who need to spin up temporary test environments on demand, and system admins managing dozens of remote clusters.

DevOps Engineer

Running status checks across multiple development or staging endpoints without opening a browser tab.

System Administrator

Adding and managing entirely new remote Docker/Kubernetes environments from the chat interface.

Software Developer

Quickly deploying a test version of an application or service into a container for immediate testing.

What Changes When You Connect

01 You manage multiple remote clusters from one spot. The `add_endpoint` tool lets you connect to a staging cluster in AWS, and then immediately check its status alongside your local development container list.

02 Eliminate manual setup headaches. Use the `init_admin` tool right away on fresh installs so your agent can grab secure credentials and start working instantly.

03 Debugging is faster than ever. Instead of logging into a dashboard just to see what's wrong, ask your agent to `list_docker_containers` and get the status in three seconds.

04 Deployment becomes conversational. You tell your agent to create a service using `create_docker_container`, specifying the image and configuration parameters directly in the chat prompt.

05 Keep services online with zero effort. If an application container stops unexpectedly, you just ask your agent to `start_docker_container` instead of writing a restart command.

Real-World Applications

Investigating Production Failures

An engineer notices latency spikes. They prompt their agent: 'List all containers in the production endpoint and tell me which ones are stopped.' The agent uses `list_docker_containers` to immediately pinpoint a misconfigured service that needs restarting.

Spinning Up Local Testing Environments

A developer needs a fresh copy of their API backend for testing. They ask the agent to deploy it using `create_docker_container`, specifying the required image and port settings, all from within their IDE chat window.

Onboarding New Infrastructure

The SysAdmin gets a new cluster endpoint URL. Instead of following a manual setup guide, they use `add_endpoint` through the agent, linking the whole remote system into their existing operational flow.

Recovering a Forgotten Service Instance

A critical caching service was accidentally stopped during maintenance. The administrator simply asks the agent to 'start the redis container in staging,' which executes `start_docker_container` instantly, restoring service.

Patterns to Avoid

Assuming connectivity

✗ AVOID

Telling the agent to list containers or start a service without first connecting it to the remote host. The operation will fail and waste time.

✓ INSTEAD

Always run `add_endpoint` first, pointing your MCP at the desired cluster URL. Once connected, you can then confidently use tools like `list_docker_containers`.

Forgetting credentials

✗ AVOID

Attempting any operation after an initial setup without securing the connection. The agent will reject the request because it lacks a valid token.

✓ INSTEAD

After setting up the endpoint, run `authenticate` to receive your JWT token. Your agent uses this token for every subsequent call.

Ignoring initial setup

✗ AVOID

Trying to use advanced features like deploying containers with custom JSON configurations without initializing admin rights first. The process will halt at a permission barrier.

✓ INSTEAD

If you're setting up Portainer for the first time, run `init_admin` immediately. This ensures your agent has the necessary starting credentials.

The Right Fit

Use this MCP if managing container lifecycle—listing, creating, or restarting services across diverse environments (Docker/K8s)—is a frequent task that involves multiple manual steps. It excels when you need to interact with infrastructure state conversationally.

Don't use it if your primary goal is complex CI/CD pipeline execution based on git hooks; dedicated pipeline tools are better for those workflows. Also, don't rely on this MCP for generating massive amounts of deployment YAML or writing application code—use a specialized coding agent for that. However, if you need to validate the *state* of containers after they've been deployed by another tool, Portainer is exactly what you need.

The Container Dashboard Click Fatigue

Right now, checking your infrastructure state involves a painful dance: logging into the staging dashboard, clicking through to the specific environment, finding the container list, and then if something's wrong, opening a second tab to figure out the restart command. It’s always clicks, tabs, copy-pasting—a massive context switch every time you need status.

With this MCP, all of that vanishes. You simply tell your agent what needs checking. The agent handles the connection, navigates the endpoints, and gives you a clean summary right where you are chatting. It takes hours of manual clicking down to a single conversation.

Portainer MCP: Container Ops in Conversation

You no longer need to remember the specific API calls or sequence of commands required for different environments. The agent abstracts that complexity away, letting you ask simple questions like, 'What's wrong with my test service?' and getting a direct answer.

It’s about control without friction. You gain total operational visibility over your entire container stack, making infrastructure management as natural as sending an email.

Portainer: 6 Tools for Container Ops

Use these tools to connect your AI agent to your container infrastructure, allowing you to manage environments, deploy services, and check status without leaving your conversation.

#	TOOL	DESCRIPTION
01	`add_endpoint`	Connects the MCP to a new environment, like a remote Docker host or Kubernetes cluster.
02	`authenticate`	Generates a temporary security token needed for all subsequent operations on the platform.
03	`create_docker_container`	Builds and creates a new container instance using a specified image name.
04	`init_admin`	Sets up the initial administrative password for a brand-new Portainer installation.
05	`list_docker_containers`	Retrieves and lists all existing containers, showing their current running or stopped status in an environment.
06	`start_docker_container`	Restarts a container that is currently stopped or offline within the specified endpoint.

See It in Action

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

U List all containers in Portainer endpoint 1.

I've retrieved the container list for endpoint 1. You have 4 containers: 'nginx-proxy' (running), 'db-postgres' (running), 'redis-cache' (stopped), and 'app-worker' (running).

U Create a new container named 'web-server' using the 'nginx:latest' image in endpoint 2.

The container 'web-server' has been successfully created in endpoint 2 using the nginx:latest image. Would you like me to start it now?

U Start the container 'redis-cache' in endpoint 1.

Starting 'redis-cache' in endpoint 1... The container has been started successfully.

Frequently Asked Questions

01 **How do I list containers using the Portainer MCP?**

You use the `list_docker_containers` tool. Just ask your agent to check a specific environment, and it will return a structured list showing all running and stopped services.

02 **Can I connect multiple remote clusters with Portainer MCP?**

Yes. You use the `add_endpoint` tool to link various local or cloud-based Docker/Kubernetes environments, allowing your agent to treat them all as one resource pool.

03 **What if a container is stopped? How do I restart it?**

Use the `start_docker_container` tool. You just need to give the agent the name of the service and the endpoint, and it handles bringing the instance back online.

04 **Do I have to set up Portainer manually before using the MCP?**

Yes. For initial setup on a new system, you must first run `init_admin` to create the administrative credentials needed for the agent to gain access.

05 **Can I deploy a brand-new container with Portainer MCP?**

Absolutely. Use the `create_docker_container` tool by telling your agent the exact image and name you want, and it handles deploying the service for you.

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 `"portainer": { "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

Portainer 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 Portainer. 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	Portainer MCP
Server ID	`019e38d9-6451-70e1-93ad-bf21ffd80161`
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/portainer.