MCP SERVER | NO CODE | CLOUD HOSTED

Dokku MCP for AI Agents

Manage Self-Hosted Container Deployment and Scaling Metrics

Dokku MCP lets you manage self-hosted applications and containers using natural language commands. Stop logging into dashboards or SSHing just to check a service status. Connect your AI agent to this MCP and get full control over your PaaS environment, allowing you to scale processes, audit environment variables, deploy updates, and stream live logs entirely through conversation.

A+ Quality Score 98.33/100

container-orchestration | paas | self-hosted | deployment | server-management | environment-variables

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

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.

Dokku MCP

10 tools available
Cloud-hosted on Vinkius

Managing self-hosted infrastructure can be a nightmare of CLI commands, dashboard clicks, and forgotten credentials. This connector hands you back the command line's power without forcing you to learn it. By connecting your AI agent to this MCP, you gain conversational control over everything running on your private PaaS. You can list every deployed application or dismantle an entire setup instantly, all while talking to your AI client. It means getting accurate environment variables and SQL credentials without ever logging into the underlying VM. Need to scale up web workers during a traffic spike? Ask for it. Run a one-off database migration script safely in isolation? Done. The Vinkius catalog makes connecting this kind of deep infrastructure control easy, allowing you to manage your entire application lifecycle from a single chat window.

Core Capabilities

01 — List all deployed applications

Retrieves an inventory of every container and service currently managed by your self-hosted Dokku instance.

02 — Deploy or remove services

Instantly provisions new application boundaries, or completely tears down existing containers and their DNS records in one conversation.

03 — Check environment settings

Reads the exact runtime configuration variables (like database keys or API tokens) bound to a specific running app.

04 — Update application configuration

Injects new sensitive environment variables and forces a rolling deployment across your cluster to ensure the changes take effect immediately.

05 — Adjust container scaling

Directly manipulates replica counts, telling the system whether web frontends or background worker tasks need to spin up or down.

06 — Stream live logs and run commands

Pulls real-time execution tails for debugging request stacks, or launches raw, isolated containers for maintenance scripts like database migrations.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to the Dokku MCP on Vinkius and provide your specific Dokku Host URL and API Token.

02 Next, connect this MCP to your preferred AI client. The agent authenticates with your private cloud instance.

03 Finally, you interact by asking natural language questions, and the agent executes the necessary infrastructure commands against your self-hosted PaaS.

The bottom line is that it turns complex terminal workflows into simple, conversational requests.

Built For

This MCP is for engineers and developers who run their own infrastructure. If your job involves deployment cycles, checking logs when things break, or managing environment variables across multiple servers, this saves you hours of manual work.

DevOps Engineer

You use it to monitor process scales and application health across multiple nodes without logging into individual machines. You can check container status or trigger restarts when load balancers fail.

Full-stack Developer

You use it to run database migrations, scale web processes for peak traffic, or inject new API keys right from the chat window instead of following a multi-step CI/CD process.

System Administrator

You use it to audit environment variables and pull system logs on demand. This lets you verify application settings or debug crashes without ever needing manual SSH access.

What Changes When You Connect

01 Eliminate repetitive CLI work. Instead of running `dokku list` or checking dashboards, you just ask your agent to 'list all apps' via the `list_apps` tool.

02 Achieve true zero-downtime updates. When you change a key setting using `set_config`, the MCP handles the rolling deployment automatically, keeping your service live while it updates.

03 Deep debugging without SSH access. The `get_logs` tool pulls detailed system execution tails and backtraces directly into the chat for rapid troubleshooting.

04 Control resources precisely. You can tell the agent to scale specific containers using `ps_scale`, ensuring workers spool up exactly when traffic demands it, saving compute costs.

05 Safe credential handling. Need to audit an API key? Use `list_config` to see the environment variables without ever having to manually access sensitive files or secrets managers.

Real-World Applications

Debugging a slow endpoint

The agent finds that the 'api-gateway' is timing out. You ask it for logs, and the MCP uses `get_logs` to pull precise system execution tails showing slow SQL queries inside the containers. The bottleneck is immediately obvious.

Rolling out a database migration

A new feature requires an updated schema. Instead of manually running a script, you instruct the agent to use `run_command` for a one-off execution of 'rake db:migrate' in an isolated container, ensuring the main app isn't interrupted.

Scaling up for peak traffic

It's Black Friday. The web process is struggling. You ask the agent to scale the web worker to five replicas using `ps_scale`. The MCP handles the load balancing update automatically, giving you immediate relief.

Revoking an expired token

An API key expires and needs replacement. Instead of manually finding the config file, you tell the agent to use `unset_config` for the old key, then immediately use `set_config` with the new secret, triggering a safe redeployment.

Patterns to Avoid

Assuming simple restarts fix everything

✗ AVOID

The app is behaving weirdly, so I just run 'restart all'. This often fails to fully cycle the process or might cause a brief outage.

✓ INSTEAD

To safely bounce an application container and ensure zero downtime deployment, use the `ps_restart` tool. This manages the port allocation across multiple replicas automatically.

Mixing up config keys

✗ AVOID

I just changed a secret key but I'm not sure if it applied. I manually check the old files, which are often out of sync with the running container.

✓ INSTEAD

Always verify your settings by asking the agent to `list_config`. This pulls the actual environment dictionary loaded into the app at runtime, giving you definitive proof.

Forgetting cleanup steps

✗ AVOID

I tested a feature and created an entire test stack, but now I'm stuck with three unused containers cluttering my host.

✓ INSTEAD

Don't leave orphaned resources. Use the `destroy_app` tool to dismantle an app completely, removing all bound containers, volumes, and DNS records in one go.

The Right Fit

Use this MCP if your job requires constant interaction with self-hosted infrastructure details: checking logs, modifying environment variables, or scaling resources. You must have direct access to a Dokku instance's REST API for it to work. Don't use it if you only need simple messaging integrations; those are better handled by generic communication tools. Also, don't attempt to use this MCP to manage cloud providers like AWS or Azure—it is strictly for self-hosted PaaS environments. If your workflow requires running a single, isolated command without affecting the live web traffic (e.g., generating a report), then `run_command` is the perfect tool; otherwise, stick to `ps_scale` when scaling processes.

Dokku MCP for AI Agents: Managing Self-Hosted Container Deployments

Today, managing a self-hosted application means jumping through hoops. You're logging into the dashboard to check container status, then switching to SSH just to view the logs, and finally running specific CLI commands for deployments or variable changes. It’s tedious, slow, and prone to copy/paste errors.

With this MCP, you talk to your agent instead. Instead of navigating complex UIs, you simply tell it what needs doing. You get conversational control over listing apps with `list_apps`, checking logs with `get_logs`, or scaling containers using `ps_scale`. The outcome is an instant command execution that updates your infrastructure state without the friction.

Dokku MCP for AI Agents: Auditing Environment Variables and Scaling

Manually auditing your application's runtime configuration means digging through multiple files or relying on memory to know which API tokens are active. It’s a security risk just waiting to happen.

This MCP solves that by letting you `list_config` securely and instantly, showing the exact environment variables loaded into the app at any moment. You can also use `set_config` to update

secrets and trigger a full deployment cycle—all from your chat interface.

Dokku: 10 Tools for Self-Hosted Infrastructure Management

Use these tools to manage application deployment, scaling events, environment variable auditing, and live log retrieval on your self-hosted Dokku instance.

#	TOOL	DESCRIPTION
01	`create_app`	Sets up the basic network structure needed for a new self-hosted application boundary on your Dokku VM.
02	`destroy_app`	Deletes an entire app, including all running containers and associated DNS routing records.
03	`get_logs`	Streams the live standard output and error logs from your application's closed Docker containers for debugging.
04	`list_apps`	Retrieves a list of all self-hosted applications currently deployed and managed by Dokku Core.
05	`list_config`	Extracts the specific environment variables loaded into an application, such as SQL credentials or external API tokens.
06	`ps_restart`	Forces a dynamic restart of a running application container to apply changes while maintaining zero downtime for other replicas.
07	`ps_scale`	Changes the structural replica count, determining if web frontends or background worker tasks should increase or decrease.
08	`run_command`	Boots a temporary isolated container to execute single commands safely, perfect for maintenance tasks like running scripts.
09	`set_config`	Injects new environment variables into an active application and triggers the mandatory deployment cycle to use them.
10	`unset_config`	Removes sensitive configuration values from the app, forcing a replacement cycle to ensure they are no longer accessible.

See It in Action

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

U List all my self-hosted apps and check if anything needs attention.

Dokku App Inventory

- `api-gateway` : Status: Running | Scale: 3/3 (Web) | Last Deployment: 2 hours ago
- `worker-queue` : Status: Running | Scale: 1/1 (Worker)
- `frontend-web` : Status: Down | Scale: 0/1 (Web)

Action Required: The `frontend-web` container is currently down. Would you like me to restart it, or should I check the logs first using the `get_logs` tool?

U I need to update my database credentials for the main API service.

Configuration Update Required

To update the credentials for `api-gateway`, we must use a multi-step process:

1. Run `unset_config` on the old variable (`DATABASE_PASSWORD`).
2. Provide the new, secure credential.
3. I will then execute `set_config` and trigger the rolling deployment to ensure zero downtime.

U Can you make sure my worker processes are scaled up for a big data job?

Scaling Request Accepted

The current web process scale is 2. I've received your request to boost the workers. Running `ps_scale` now to set the replica count for 'worker-queue' to 10.

Status: Scaling initiated... ⏳

I will notify you when the internal load balancer confirms all 10 worker endpoints are active and ready.

Frequently Asked Questions

01 **How do I manage my self-hosted containers using the Dokku MCP for AI Agents?**

You interact with your infrastructure by simply asking questions. You can ask to scale processes, check logs, or list deployments—it converts complex terminal commands into conversational requests.

02 **Does connecting the Dokku MCP for AI Agents mean I don't need SSH access anymore?**

Not usually. This MCP allows your agent to perform crucial tasks like logging and scaling without you needing manual SSH access, automating those routine checks into a chat interface.

03 **What if I need to run a one-off migration script for my app?**

You can use the MCP's capabilities to launch raw commands inside temporary, isolated containers. This lets you safely run scripts like migrations without affecting your live web traffic.

04 **Can I update sensitive environment variables with this Dokku MCP for AI Agents?**

Yes. You can inject or remove secrets and credentials using the MCP tools, which then forces a rolling deployment to ensure the change is applied safely across all your running services.

05 **Is this only for web apps, or can it handle background workers too?**

It handles everything. You can monitor and scale both the front-facing web containers and dedicated worker background tasks using specific tools like `ps_scale` to match demand.

06 **How does the Dokku MCP for AI Agents handle deployment updates?**

It manages it automatically. When you update a configuration or code, the MCP triggers a controlled rolling deployment cycle that minimizes downtime and ensures all replicas get the new version safely.

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 `"dokku": { "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

Dokku 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 Dokku. 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	Dokku MCP
Server ID	`019d7588-3e17-709b-a1af-34501221def5`
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/dokku.