MCP SERVER | NO CODE | CLOUD HOSTED

Baota Panel MCP for AI Agents

Manage web hosting resources and system performance metrics

Baota Panel / 宝塔面板 API connects your AI agent directly to one of China's leading web hosting control panels. It lets you manage complex server infrastructure—from listing all active sites and checking system load to auditing databases and viewing logs—all through conversation. You get instant, deep insight into your entire stack without ever touching the panel interface.

A+ Quality Score 100/100

server-management | web-hosting | database-monitoring | system-resources | control-panel

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/baota-panel-api

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.

Baota Panel / 宝塔面板 API MCP

10 tools available
Cloud-hosted on Vinkius

This MCP gives your AI agent full access to managing critical server components managed by Baota Panel. Think of it as giving your chat client a virtual set of hands that can log in and run every command, but you talk to it naturally.

Instead of clicking through endless dashboards to check if the CPU spiked or if a site is running out of disk space, you just ask. Your agent pulls real-time data: it lists all managed websites, checks current system load against usage limits, and even provides metadata for every database your company runs. This means security audits, performance reviews, and routine maintenance happen in seconds.

When you connect Baota to Vinkius through any MCP-compatible client, your agent acts like a dedicated Site Reliability Engineer (SRE), keeping your infrastructure accurate and running smoothly 24/7. It's about moving from manual checks across multiple tabs to one conversational query.

Core Capabilities

01 — Audit System Health

Get real-time metrics on the server, including disk usage, network status, and overall system load.

02 — Manage Web Assets

List all websites hosted on the panel and get detailed information about each site's configuration.

03 — Monitor Data Stores

Retrieve a list of all databases, plus metadata for your entire data storage infrastructure.

04 — Review Operational History

Browse administrative logs and view pending background tasks to understand system activity.

05 — Inventory Software

List all installed software and plugins, such as Nginx, PHP, or MySQL, for quick inventory checks.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/baota-panel-api** — connect your AI agent in three steps.

01 Subscribe to this MCP on Vinkius. You'll need your Panel Host, API Secret Key, and API ID.

02 Enter those credentials into your AI client. Make sure the calling IP address is whitelisted inside the Baota Panel settings for access.

03 Start chatting with your agent. Ask it specific questions about your server resources or managed sites to see the data come back.

The bottom line is, you pass credentials once, and then your AI client handles all the connection details so you can manage your entire web stack through natural language conversation.

Built For

This MCP is for Ops Engineers who hate clicking through resource dashboards at 2 AM. It's built for Web Developers who need to audit database lists or site configs fast, and System Admins needing a single window to check everything from logs to CPU usage.

DevOps Engineer

Uses the MCP to monitor cron jobs and network traffic status across multiple environments without logging into individual servers.

System Administrator

Runs system health checks, querying disk usage and CPU load in a single conversational prompt for compliance reports.

Web Developer

Checks site configurations or retrieves database metadata to confirm credentials before deploying new code.

What Changes When You Connect

01 Immediate resource checks: Instead of logging in to check disk usage or CPU load, you simply ask your agent. It immediately gives you the data using the `get_disk_info` tool.

02 Full site overview: You can instantly list all managed websites and get key details for each one using the `list_sites` function. This is huge for quick audits.

03 Database visibility: Need to know what databases exist? The agent lists them out using `list_databases`, giving you a full inventory without manual exploration.

04 Operational transparency: Review administrative logs or scheduled tasks by calling `list_logs` or checking cron jobs with `list_cron_tasks`. You never miss an audit trail.

05 Unified dashboard access: This MCP consolidates checks—system load, network status, and software versions—into one conversational interface via tools like `get_system_total`.

Real-World Applications

Emergency Performance Audit

The Ops Engineer notices slow site speed. Instead of manually SSHing in and running multiple commands, they ask their agent to check the system load (`get_system_total`), disk usage (`get_disk_info`), and network status (`get_network_info`). The agent compiles a single report confirming resource exhaustion.

Pre-Migration Site Inventory

The Web Developer needs to move a site, so they ask the agent to use `list_sites` first. It provides a list of all active domains and their basic configurations, ensuring nothing gets missed during the transfer.

Security Compliance Check

The System Admin runs an audit by asking for recent administrative logs (`list_logs`) and checking pending background tasks (`get_task_count`). This confirms no unauthorized access or unusual activity occurred overnight.

Platform Maintenance Planning

Before a major update, the team member asks the agent to use `get_software_list` and `list_databases`. They get an immediate inventory of all installed components (Nginx, PHP versions) and data stores needing attention.

Patterns to Avoid

Trying to list everything manually

✗ AVOID

Manually navigating the Baota Panel web interface through multiple sections just to gather a comprehensive list of sites, databases, and logs. This takes 20 minutes.

✓ INSTEAD

Use your agent with this MCP. Just ask it to 'list all managed websites' or 'show me the database inventory.' The agent handles the clicks using `list_sites` and `list_databases`, giving you a clean list instantly.

Ignoring resource metrics

✗ AVOID

Assuming the server is healthy because no errors are visible, but missing signs of slow degradation like creeping disk usage or high load averages.

✓ INSTEAD

Run system health checks by asking for `get_disk_info` and `get_system_total`. This gives you hard metrics on resource capacity that tell you if the server is nearing its breaking point.

Missing scheduled tasks

✗ AVOID

A site breaks because a critical cron job failed, but no one noticed until users complained. The logs were too deep to check manually.

✓ INSTEAD

Ask your agent to use `list_cron_tasks`. This shows you exactly what is scheduled to run and if there are any obvious gaps or outdated jobs.

The Right Fit

Use this MCP when the core problem is gathering deep, diverse information about a web hosting stack. If your job involves regularly checking server health, inventorying assets (sites, databases), or auditing logs across multiple sub-systems, this is for you. Don't use it if you just need to deploy a single piece of code; for that, a pure CI/CD tool is better. Also, don't try to *change* settings—this MCP is designed for read-only monitoring and listing. If you need to perform write operations like deleting a user or changing a configuration setting, look for an MCP with explicit modification tools.

Baota Panel / 宝塔面板 API: Auditing Web Hosting Resources

Right now, checking if your server is healthy means opening the control panel, jumping between tabs for disk usage, running separate checks for network status, and then manually listing sites to verify their existence. It's a tedious cycle of clicking through multiple dashboards just to get a unified view.

With this MCP, you talk to your agent once. You simply ask it to 'Give me the full server health report.' The agent uses its tools—like checking system load and retrieving disk info—and gives you one complete answer instantly. It's immediate, comprehensive insight.

Baota Panel / 宝塔面板 API: Managing Databases and Sites

Finding all the databases or checking which sites are active is usually a deep dive into separate sections of the panel. You have to remember whether the information was under 'Databases' or 'Websites,' forcing you to manually cross-reference multiple areas.

Now, your agent handles it. A single request retrieves both the database inventory and the site list—using tools like `list_databases` and `list_sites`. You get a clean, actionable report covering all web assets in one go.

Baota Panel: 10 Tools for Web Hosting Resource Monitoring

Use these tools to gather disk usage, network status, software lists, system load metrics, cron jobs, databases, FTP accounts, logs, and sites.

#	TOOL	DESCRIPTION
01	`get_disk_info`	Retrieves the current disk usage statistics for the managed server.
02	`get_network_info`	Pulls the real-time status and metrics of the server's network connection.
03	`get_software_list`	Lists all software, plugins, and versions installed on the control panel.
04	`get_system_total`	Provides detailed information about the server's overall system load and performance metrics.
05	`get_task_count`	Checks how many background tasks are currently pending or waiting to run.
06	`list_cron_tasks`	Lists all scheduled cron jobs and their associated parameters on the panel.
07	`list_databases`	Provides a comprehensive list of every database managed by the hosting panel.
08	`list_ftp`	Lists all configured FTP accounts and their associated usernames.
09	`list_logs`	Retrieves recent administrative logs for auditing system activity.
10	`list_sites`	Generates a list of all active and managed websites under the panel.

See It in Action

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

U What's wrong with the server? Check everything from disk space to network status.

System Health Report: [Date/Time]

- 💾 **Disk Usage:** 78% used (Limit: 100%). Action needed on `/var/www`.
- 🌐 **Network Status:** Operational. Latency average: 45ms.
- 📈 **System Load:** Average load is 2.1/4 cores. Peak usage observed at 9 AM.

Overall, disk space is the main concern here.

U List all my websites and tell me what plugins or software versions they use.

Active Site Inventory

SITE NAME	STATUS	PRIMARY SOFTWARE	KEY PLUGINS
marketing.com	Active	Nginx 1.25	WordPress, Yoast SEO
api-server.net	Active	PHP 8.1	Custom API Gateway

You can also check the full list of installed software using a separate query.

U Show me any recent security or admin activity that happened today.

Recent Administrative Logs (Last 5 Entries)

- **[10:30 AM]**: User 'admin' modified firewall rules.
- **[9:15 AM]**: Database backup completed for `user_db`.
- **[8:00 AM]**: Site `staging.com` created successfully.

No suspicious activity detected in the visible logs.

Frequently Asked Questions

01 **How do I use Baota Panel / 宝塔面板 API to check server CPU usage?**

You just ask your agent for 'system resource usage.' It runs a background check and returns the current load average, CPU percentage, and memory stats. This saves you from having to manually run monitoring commands.

02 **Can Baota Panel / 宝塔面板 API list all my domains?**

Yes, it can. By asking your agent to 'list managed sites,' it pulls a complete inventory of every website hosted on the panel. This is perfect for audits or migration planning.

03 **I need help checking database health with Baota Panel / 宝塔面板 API.**

Your agent can list all databases and provide metadata, confirming if they exist and what kind of data they hold. This gives you a quick overview without needing direct access to the database console.

04 **What is the easiest way to check system logs using Baota Panel / 宝塔面板 API?**

Ask your agent for 'recent administrative logs.' It pulls and summarizes the most critical entries, letting you know what happened on the server without having to scroll through thousands of lines of text.

05 **Does Baota Panel / 宝塔面板 API help me manage web hosting credentials?**

It helps with visibility. You can list all configured FTP accounts and even see which software stacks (like Nginx or MySQL) are installed, giving you a full picture of your assets.

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 `"baota-panel-api": { "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

Baota Panel / 宝塔面板 API 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 Baota Panel / 宝塔面板 API. 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	Baota Panel / 宝塔面板 API MCP
Server ID	`019d841c-9bfb-709b-a6aa-7e86c1efd1c7`
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/baota-panel-api.