MCP SERVER | NO CODE | CLOUD HOSTED

Paperspace MCP

See GPU status, deployments, and compute resources instantly.

Paperspace MCP gives your AI client visibility into complex cloud machine learning environments. Use it to list active compute instances, trace deployed services, inspect Jupyter notebooks, and map user accounts across deep learning infrastructure. It's essential for anyone needing real-time status on GPU resources.

A+ Quality Score 100/100

gpu-provisioning | machine-learning | cloud-computing | jupyter-notebooks | container-management | infrastructure-monitoring

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

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.

Paperspace MCP

6 tools available
Cloud-hosted on Vinkius

Managing distributed computing power is a headache until now. This MCP connects your AI agent directly to Paperspace Cloud Insights, giving you an immediate view of every active resource running in the cloud. You can query which physical machine cores are heavily modified or check memory schemas across different compute instances. It's also great for auditing who has access by checking native identity accounts and tracking team project limits. If your workflow requires knowing the status of deployed containers, this MCP handles that too. This level of deep infrastructure insight is what makes the Vinkius catalog so powerful; you get one connection point to dozens of specialized services. You can use it to inspect raw Jupyter notebooks linked to specific deep learning models or even check if a serverless API container is available by reviewing its logs.

Core Capabilities

01 — Check active compute resources

Identify all provisioned machine cores, checking their current status and resource limits.

02 — Audit deployed services

Retrieve logs and statuses for specific cloud deployment targets to ensure containers are available.

03 — Inspect ML project boundaries

List structured project groupings, verifying team limits and GPU unit assignments across the platform.

04 — Query notebook usage

Find details on Jupyter notebooks by inspecting deep internal arrays that govern AI workloads.

05 — Verify user identity access

Identify all linked account identities and associated billing or support plan constraints.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP in the Vinkius catalog.

02 Provide your Paperspace API Key to your AI client.

03 Use natural language prompts with your agent to monitor specific GPU footprints or deployment logs.

The bottom line is, you tell your AI client what infrastructure detail you need, and it executes the query directly against Paperspace.

Built For

This MCP is for ML Infrastructure Engineers, Data Scientists, and DevOps Ops who spend too much time clicking through different vendor dashboards to check resource status. If your job involves tracking which GPU cluster is overloaded or verifying a deployed container's health, you need this.

ML Operations Engineer

Uses the MCP to confirm that newly trained models have stable compute resources and checks for active deployment targets.

Data Scientist

Needs to verify memory constraints or review raw Jupyter notebooks attached to a specific research project before presenting results.

Infrastructure Architect

Audits the entire environment by listing active machines and tracing team projects to ensure resource boundaries are respected.

What Changes When You Connect

01 Get a precise overview of your hardware. Instead of logging into the console to check available resources, use `list_machines` to get an instant count of all bounded compute instances.

02 Audit resource usage quickly. You can run `get_machine_details` on any instance to extract specific properties like memory schemas and storage constraints without manually inspecting dashboards.

03 Track team work accurately. Use `list_projects` to see exactly how GPU units are grouped into discrete projects, making it easy to audit team limits.

04 Verify container health automatically. Checking deployment logs via `list_deployments` tells you immediately if your serverless API containers are currently active and running correctly.

05 Deep dive into research code. If a notebook is behaving strangely, use `list_notebooks` to inspect the underlying deep arrays governing that specific AI workload.

Real-World Applications

Debugging an Overloaded Cluster

A data scientist notices slow model performance. They prompt their agent: 'List all machines and check which ones are hitting memory limits.' The agent uses `list_machines` combined with `get_machine_details` to pinpoint the exact overloaded GPU instance, allowing immediate scaling decisions.

Auditing Team Access

An infrastructure architect needs to confirm who has access before a major migration. They ask their agent to run `get_user_details` and review all active team limits using `list_projects`, ensuring no unauthorized accounts exist.

Checking Production Readiness

A DevOps engineer needs to verify if the latest API build is ready. They ask their agent to run `list_deployments` to check container logs, confirming that all required services are reporting 'available' status.

Investigating Stalled Research

A researcher suspects a Jupyter notebook has crashed without logging off. They prompt their agent to run `list_notebooks`, which retrieves the deep internal arrays and confirms if the workload is still provisioned or stalled.

Patterns to Avoid

Checking status via UI clicks

✗ AVOID

Manually logging into Paperspace, navigating to the Compute section, clicking through multiple tabs, and copying down resource IDs takes 15 minutes of wasted time.

✓ INSTEAD

Just ask your agent. Use `list_machines` or `get_machine_details`. Your AI client handles all the navigation and data extraction in seconds.

Guessing deployment targets

✗ AVOID

Assuming a container is running because it was deployed last week, without confirming its current operational status.

✓ INSTEAD

Always use `list_deployments`. This tool reads the actual cloud logs to confirm if the container is currently reporting an active and available state.

The Right Fit

Use this MCP if your workflow involves querying the *state* of compute infrastructure—specifically GPU allocation, deployed service health, or project-level resource boundaries. It's built for ML Ops and Infra teams who need to know 'what is running, where, and how big.'

Don't use this if you simply need to write code (use a general coding assistant) or manage user billing outside of basic identity verification (`get_user_details`). If your goal is general database interaction or CRM updates, look for an MCP specific to those domains. This MCP is purely about hardware and deployment lifecycle monitoring.

The Manual Headache of Cloud Resource Auditing

Right now, figuring out the status of your compute environment means jumping between a dashboard, running separate CLI commands for

With this MCP integrated into Vinkius, the entire process collapses into a single chat prompt. You ask your agent what's running—whether it's an

deployments, and manually checking notebook IDs. You're copying resource IDs from one screen to another just to confirm if that GPU is actually free or if that container failed overnight. It's tedious, slow, and you always feel like you missed something.

active machine core or a Jupyter notebook limit—and it delivers a clean, consolidated report instantly.

Paperspace MCP: Instant Infrastructure Visibility

You no longer have to manually check individual team projects or cross-reference machine IDs with project groupings. The agent uses `list_projects` and `get_machine_details` together, giving you a single view of the entire resource map.

The difference is control. Your AI client doesn't just read data; it structures complex cloud metadata into actionable intelligence in seconds.

Paperspace: 6 Tools for Infrastructure Management

Use these tools to get deep reads on your cloud infrastructure, from listing machine IDs to checking deployment logs.

#	TOOL	DESCRIPTION
01	`list_machines`	Lists all bounded compute resources available within your Paperspace account limits.
02	`get_machine_details`	Extracts detailed properties for a specific machine instance, including its current logic status.
03	`list_deployments`	Retrieves explicit logs and statuses for cloud deployment targets.
04	`list_notebooks`	Inspects deep internal arrays to find details about specific AI workload notebooks.
05	`list_projects`	Enumerate structured project groupings, showing which team limits are currently active.
06	`get_user_details`	Identifies precise account details and associated authentication arrays for the user.

See It in Action

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

U Scan Paperspace for any currently active deployed Core machines.

Successfully queried physical limit layers. Found 2 virtual Core instances. Instance `ps1aaq4`: P4000 GPU (Off). Instance `ps38jxx`: RTX5000 GPU (Running) allocated 30GB of internal RAM. Should I describe `ps38jxx` deeply?

U Execute an inventory sweep over active Gradient Jupyter Notebooks running in production.

Tapped native AI layer arrays determining 1 Gradient environment running explicitly in the background. Token id `nxxx`. Attached to deep learning target container environment. Status: Provisioned. The workspace maps exclusively back to 'Computer Vision Team 1'.

U Show exactly which users are tied down to my native Paperspace environment.

Parsed global arrays confirming absolute account identities. Account belongs to "Team Vinkius". Billing Profile: Active Card. Support Plan: Developer. Storage constraint ceiling set to strictly 2000 GB.

Frequently Asked Questions

01 **How does Paperspace MCP help me find an idle GPU?**

You can use `list_machines` to see all available compute resources. Then, prompt your agent to run `get_machine_details` on those IDs to check their current load and memory usage.

02 **Can I track which team owns a specific project using Paperspace MCP?**

Yes, running `list_projects` enumerates all structured groupings. This tool shows the active team limits attached to specific GPU units.

03 **What if my container deployment log is corrupted? Can Paperspace MCP help?**

You can use `list_deployments`. The MCP reads explicit cloud logs, which helps verify whether the target deployment status remains active even if other logging methods fail.

04 **Does Paperspace MCP only work for new ML projects?**

No. It monitors existing infrastructure too. Use `list_notebooks` to inspect old or dormant Jupyter notebooks and check their associated workload limits.

05 **How do I know which user account is connected to this Paperspace MCP?**

Running the `get_user_details` tool identifies all active account arrays, confirming who has access credentials and what support plan they are under.

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 `"paperspace": { "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

Paperspace 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 Paperspace. 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	Paperspace MCP
Server ID	`019d75ee-db8c-73a3-b9bb-0ca4e354d0d4`
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/paperspace.