MCP SERVER | NO CODE | CLOUD HOSTED

Lambda Labs (GPU Cloud) MCP

Manage your entire GPU cluster via natural conversation.

Lambda Labs (GPU Cloud) MCP connects your AI client directly to high-performance GPU infrastructure. Use natural conversation to launch H100 or A100 virtual machines, monitor ML workloads, check pricing, and manage secure SSH keys without touching a dashboard.

A+ Quality Score 100/100

gpu-cloud | machine-learning | infrastructure-as-code | virtual-machines | ai-training | ssh-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/lambda-labs-gpu-cloud

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Lambda Labs (GPU Cloud) MCP

7 tools available
Cloud-hosted on Vinkius

This MCP gives you full control over powerful cloud compute resources through conversation. Instead of logging into a separate web portal and clicking through menus to provision hardware, your agent handles the entire workflow. You can ask it to launch specific GPU types for training or fine-tuning —say, an H100 box in us-east-1. Need to check which shared file systems are available across multiple workers? Just ask. If you need to shut down a running job to stop billing immediately, the agent terminates it instantly. It also keeps track of all your globally managed SSH keys and helps map persistent storage volumes for multi-node setups. When you connect this MCP through Vinkius, your AI client becomes an infrastructure expert, making complex resource management feel like chatting with a teammate.

Core Capabilities

01 — Provisioning Compute Resources

Launch new GPU virtual machines (H100/A100) and manage their entire lifecycle from start to finish.

02 — Monitoring Instance Status

List all currently running instances and retrieve key details like hardware specs, public IPs, and Jupyter Lab tokens.

03 — Inventory and Cost Planning

Discover available GPU node types across different regions and check their current pricing to plan budgets.

04 — Secure Access Management

View or manage the globally stored SSH public keys required for secure, zero-trust access over port 22.

05 — Shared Storage Mapping

Discover persistent shared NAS volumes available to mount across multiple worker nodes simultaneously.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/lambda-labs-gpu-cloud** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Lambda Labs API Key.

02 Connect the credentials to your preferred AI client (Claude, Cursor, etc.).

03 Ask your agent natural language questions like, 'Launch a 1x H100 instance in us-east-1 with key X' or 'List all running GPU instances.'

The bottom line is you use conversational prompts to manage complex infrastructure tasks that used to require manual API calls and dashboard navigation.

Built For

This MCP is for the ML Engineer who spends hours clicking between dashboards just to get a machine running. It's for the Data Scientist who needs fast, ad-hoc access to powerful compute without writing boilerplate code. If your job involves managing high-stakes GPU clusters, this saves time and headaches.

Machine Learning Engineer

Launching large GPU boxes for training or fine-tuning; checking if the required SSH keys are managed correctly.

Data Scientist

Monitoring active instances and retrieving Jupyter Lab access tokens so they can jump into rapid experimentation immediately.

AI Infrastructure Ops Specialist

Managing shared file systems across multiple worker nodes and ensuring the compute nodes are properly terminated to stop billing.

What Changes When You Connect

01 Launch powerful machines on demand. Instead of manually going through a dashboard to provision compute, you can ask the agent to launch an H100 instance instantly.

02 Stop wasting money immediately. Use the termination tool to destroy compute nodes and stop billing with just a simple command, preventing accidental charges.

03 Know your options before you start. You can use list_instance_types to discover every GPU node type and check its current pricing across various regions for budget planning.

04 Maintain secure access easily. The agent lets you manage SSH keys by listing all globally managed public keys without having to log into a separate key management system.

05 Keep your data centralized. Use list_filesystems to map shared NAS volumes, ensuring that every worker node can mount the same persistent storage for training data.

Real-World Applications

Scaling up for a large model run

A Machine Learning Engineer needs 10 A100 GPUs in us-east-1. Instead of manually checking capacity and clicking 'launch' multiple times, they ask their agent to launch the required instances. The agent handles the provisioning using launch_instance and returns a list_instances confirmation.

Auditing security access

An Ops Specialist needs to verify who has SSH access across all clusters. They use list_ssh_keys, which immediately enumerates every globally managed public key, ensuring compliance and zero-trust policies are met.

Debugging a failed job

A Data Scientist finds an instance is stuck running old code. They realize they need to stop it before the next billing cycle hits. They ask the agent to terminate_instances, which immediately stops compute and clears the resource.

Planning a multi-region deployment

A team lead needs to know if they can deploy their model training across two different geographical areas. They use list_instance_types to get the full pricing matrix and check physical availability in both regions.

Patterns to Avoid

Using basic scripting for everything

✗ AVOID

Trying to write a script that handles instance creation, key injection, status checking, and billing termination. This results in massive amounts of complex code with many failure points.

✓ INSTEAD

Use this MCP's conversational tools like launch_instance, list_instances, and terminate_instances. Your agent manages the complexity behind the scenes, letting you focus on what you're building.

Manually checking billing dashboards

✗ AVOID

Logging into 3 different cloud provider portals just to see if a compute job is still running and costing money. This wastes time and often results in missing a key status update.

✓ INSTEAD

Use list_instances to get an immediate, centralized view of all active jobs and their specs. If you're done with the job, use terminate_instances.

Copy-pasting connection strings

✗ AVOID

Getting a cluster ID from one dashboard and then manually pasting it into another system to get the SSH string. This is slow and prone to copy errors.

✓ INSTEAD

Ask the agent to use get_instance for the specific machine you need, which returns the exact details and connection string in one conversational response.

The Right Fit

Use this MCP if your primary pain point is managing cloud infrastructure resources (GPU machines, SSH keys, shared storage) through a complex web interface or rigid API calls. It's perfect for engineers who prefer talking to their tools rather than writing boilerplate provisioning scripts.

Don't use this MCP if you simply need to read static data—like viewing the documentation page for H100 specs. In that case, a simple knowledge base search is enough. Also, if your entire workflow fits into one single, repeatable Python function call without needing status checks or termination capability, then a dedicated code library might be cleaner. But if you need to orchestrate multiple steps—like checking pricing, launching the machine, and verifying the key—this MCP handles the full cycle conversationally.

The headache of managing GPU clusters today

Right now, getting a compute job running is a multi-step nightmare. You have to jump between the pricing page, the instance dashboard, and the key management console. You click to see if H100s are available; then you switch tabs to launch the machine; next, you find the correct SSH key ID, paste it in, wait for provisioning, and finally, you write down the resulting IP address so your team can connect. It’s clicking, copying, pasting—over and over.

With this MCP, that whole sequence collapses into a conversation. You simply ask your agent to launch the machine. The agent checks availability, provisions the hardware using its internal tools, handles the key injection, and gives you the ready-to-use connection details in one reply. It makes infrastructure management feel like talking to a teammate who already knows how it works.

Getting compute resources with Lambda Labs (GPU Cloud) MCP

The manual process of checking resource status and managing costs involves logging into multiple

Now, you simply ask your agent to list_instances. It gives you a real-time count of what's active and

dashboards just to find out if a job is still running or how much it cost. You might forget to terminate the node, leading to unexpected bills.

their specs. Better yet, if the work is done, asking for termination is instant. The result is immediate control; you know exactly when resources are live and when they’re gone.

Lambda Labs (GPU Cloud) MCP with 7 Tools

Use these tools to list, launch, and control every aspect of your GPU infrastructure—from individual instances to shared file systems.

#	TOOL	DESCRIPTION
01	`list_instances`	Retrieves a list of every GPU instance currently running on your Lambda Cloud account.
02	`get_instance`	Pulls detailed information and the specific SSH connection string for one chosen instance.
03	`launch_instance`	Provisions a brand-new GPU virtual machine, like an H100 box, ready for secure access over port 22.
04	`terminate_instances`	Permanently and immediately destroys running GPU instances to stop billing and clean up resources.
05	`list_instance_types`	Shows the catalog of available GPU node types, their specs, pricing, and current regional availability.
06	`list_ssh_keys`	Lists all globally managed SSH public keys within your Lambda infrastructure for auditing purposes.
07	`list_filesystems`	Maps out persistent, shared NAS volumes available for mounting across multiple compute nodes.

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 running GPU instances in Lambda Cloud

I've found 2 active instances: 1) 'training-node-01' (1x H100, Region: us-east-1, Status: active), 2) 'inference-test' (1x A100, Region: us-west-1, Status: booting). Would you like the SSH connection strings for either of these?

U Launch a 1x H100 instance in us-east-1 with my 'default-key' SSH key

Launching GPU instance... I've triggered the provision of a 1x H100 box in us-east-1 with your 'default-key' attached. The Instance ID is 'inst-98765'. It should be ready for SSH access in approximately 2 minutes.

U What are the available instance types and their current pricing?

I've retrieved the latest catalog. Highlights include: gpu_1x_h100 ($2.49/hr), gpu_8x_h100 ($19.92/hr), and gpu_1x_a100_pcie ($1.10/hr). Availability is currently highest in the us-east-1 and us-south-1 regions. Would you like to check for a specific hardware setup?

Frequently Asked Questions

01 **How do I find out what GPU types are available using Lambda Labs (GPU Cloud) MCP?**

You use list_instance_types. This tool shows you the full catalog, including hardware specifications, regional availability, and current pricing matrices so you can plan your training budget.

02 **Can I launch a new GPU machine using Lambda Labs (GPU Cloud) MCP?**

Yes, use the launch_instance tool. You tell your agent what size and type of box you need, like an H100 or A100, and it handles the provisioning process.

03 **Does list_instances show me which machine I should connect to?**

list_instances shows a current list of all active compute nodes. If you need the exact connection string for one of those machines, ask the agent to run get_instance.

04 **How do I ensure my team can access files across multiple machines?**

You use list_filesystems to map out all persistent shared NAS volumes. This ensures that data stored in one location can be mounted simultaneously by every worker node your model uses.

05 **Is terminating an instance permanent and safe?**

Yes, terminate_instances permanently destroys the GPU machine. Be careful because attached ephemeral drives are vaporized immediately, but it's the fastest way to stop billing.

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 `"lambda-labs-gpu-cloud": { "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

Lambda Labs (GPU Cloud) 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 Lambda Labs (GPU Cloud). 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	Lambda Labs (GPU Cloud) MCP
Server ID	`019d75c3-c8b8-7340-8de9-5a2f3596ff1b`
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/lambda-labs-gpu-cloud.