MCP SERVER | NO CODE | CLOUD HOSTED

H2O.ai MCP

Monitor ML models and cluster health via conversation.

H2O.ai controls your entire machine learning lifecycle directly from your AI agent. Use this MCP to audit model performance, track training jobs, and check the health of your cloud cluster without manually opening dashboards or running complex commands.

A+ Quality Score 100/100

machine-learning | model-lifecycle | data-frames | predictive-modeling | cluster-monitoring

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/h2oai

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.

H2O.ai MCP

6 tools available
Cloud-hosted on Vinkius

This connector lets you manage everything happening inside your H2O.ai instance using natural conversation. Instead of logging into a dashboard, you simply ask your agent for status updates. You can review existing machine learning models by listing them and verifying their performance metrics. Need to check the underlying data? Ask the MCP to list available structured datasets or retrieve specific columns from a frame. It even monitors long-running tasks; just query the jobs list to see if training is on track. Plus, you never have to worry about hardware limits again, because you can always ping root endpoints using `cloud_status` to verify memory utilization and overall cluster health. Connecting this MCP via Vinkius means your agent has instant access to all these deep ML operations, letting you orchestrate complex data science workflows in plain English.

Core Capabilities

01 — Audit Model Inventory

List, check details for, and verify the performance metrics of every machine learning model saved in your H2O cluster.

02 — Track Data Sources

View structured datasets loaded into the cluster or retrieve specific dimensional data columns from a frame.

03 — Monitor Training Jobs

Check the status and progress of queued or running model training jobs over time.

04 — Assess Cluster Health

Get real-time diagnostics on the physical hardware, including memory usage and operational status of the cloud cluster.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/h2oai** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your H2O.ai Base URL.

02 Connect your agent (Claude, Cursor, etc.) using the provided credentials.

03 Start asking natural language questions like 'Show me all running jobs' or 'What is the memory usage?'

The bottom line is that you get a single conversational entry point to manage complex data science operations.

Built For

This MCP targets ML Engineers and Data Scientists who spend too much time switching between dashboards, running manual checks, and verifying model versions. It's for anyone whose job requires deep visibility into a live machine learning pipeline.

ML Engineer

Audits deployment status by listing models or checking the `cloud_status` endpoint to ensure resources are available before deploying.

Data Scientist

Orchestrates complex data workflows by retrieving frames with `get_frame` and monitoring job progress using `list_jobs`.

Product Manager (AI)

Verifies the availability of core AI assets, checking if required models exist via `list_models` or if data is ready for testing.

What Changes When You Connect

01 You stop clicking through multiple dashboards to check status. With `cloud_status`, you get a single, immediate report on hardware health and memory usage for your entire H2O instance.

02 Model auditing becomes instant. Instead of manually scrolling version logs, use `list_models` or `get_model` to pull performance metrics and verify which models are deployed.

03 Data preparation is faster. Use `list_frames` to see what data is available in the cluster, then use `get_frame` if you need specific dimensional columns for a test run.

04 Tracking pipelines is straightforward. The `list_jobs` tool gives you a quick overview of all running training tasks and how far along they are.

05 The process moves from manual effort to conversation. By connecting this MCP via Vinkius, your agent handles the complex API calls behind the scenes so you just talk to it.

Real-World Applications

Verifying Pre-Deployment Data Readiness

A data scientist needs to ensure a new model can use the correct data fields. They ask their agent to `list_frames` to confirm the dataset exists, then use `get_frame` on that frame name to validate the precise column names before running training.

Debugging Failed Live Models

A product team notices a model's performance dip. They ask their agent to run `get_model` for that specific asset, which immediately returns detailed metrics and configuration blocks needed to diagnose the failure point.

Checking Resource Limits Mid-Run

An ML engineer is running a large training job. Before committing resources, they ask the agent to `cloud_status` to check current memory utilization and confirm there's enough overhead for the next task.

Auditing Historical Runs

A developer needs to know which models were trained last week. They use the MCP to `list_models`, filtering by date, then ask the agent to `list_jobs` to see the execution history for those specific model names.

Patterns to Avoid

Manually checking resource limits

✗ AVOID

Opening the cloud dashboard and scrolling through metrics tabs trying to find out how much memory is left or if a node is failing.

✓ INSTEAD

Just ask your agent for the 'cloud status'. It runs `cloud_status` and tells you the current hardware health, memory usage percentage, and operational status immediately.

Assuming data structure

✗ AVOID

Writing code that assumes a dataset has columns like 'user_id' when the actual schema is different, leading to runtime errors.

✓ INSTEAD

First, use `list_frames` to see all available datasets. Then, use `get_frame` on the correct frame name to confirm the exact column names and structure before writing any code.

Confusing job status with model version

✗ AVOID

Believing that a running job means the model is updated, when in fact the job might be using an old, unlisted asset.

✓ INSTEAD

Always check the `list_models` output to confirm the specific version you intend to use. Then, verify progress by asking about the jobs with `list_jobs`.

The Right Fit

Use this MCP if your primary need is deep visibility into a live ML development environment—specifically monitoring model performance, data availability (frames), and underlying infrastructure health. You should use it when you can describe a task like 'Check the status of my cluster' or 'List all models trained last quarter.' Don't use this if you just need basic file system access (like uploading raw CSVs) or if your goal is purely model deployment to an external, non-H2O system. For simple data retrieval without ML context, a generic database connection tool might be better.

The Pain of the Dashboard Hop

Today, checking your model's health is a nightmare. You have to jump between five different tabs: one for cluster metrics, another for job logs, and three more just to confirm what data fields were used in that last training run. It’s an exercise in copy-pasting errors and switching context every two minutes.

With this MCP, all of that complexity disappears into a single conversation thread. You ask the agent about model performance or cluster utilization, and it pulls the necessary information—from `get_model` to `cloud_status`—and presents you with one clean answer.

H2O.ai MCP Gives You Conversational ML Control

You no longer need to remember the exact API endpoint or run a complex sequence of shell commands just to verify data schemas or job statuses. The agent handles that entire backend choreography for you.

It's simple: ask your agent, 'What is the status of my cluster?' and it instantly runs `cloud_status` without you having to type anything but a question. That’s the difference.

H2O.ai: 6 Tools for ML Operations

Use these tools within your AI agent to manage the entire machine learning lifecycle—from checking data frames to auditing cluster status.

#	TOOL	DESCRIPTION
01	`list_frames`	Lists all structured datasets (frames) currently loaded into the H2O cluster.
02	`get_frame`	Retrieves specific column data from a named dataset frame.
03	`list_models`	Lists all machine learning models that have been saved and tracked in the cloud instance.
04	`get_model`	Retrieves detailed information, including performance metrics, for a specific named model.
05	`list_jobs`	Lists recent and ongoing tasks or training jobs running on the cluster.
06	`cloud_status`	Gathers hardware health metrics, including memory usage and operational status of the entire cloud cluster.

See It in Action

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

U List all machine learning models in my H2O cluster

Retrieving models... I found 5 models including 'GBM_model_1', 'DeepLearning_v2', and 'AutoML_leader'. Would you like the detailed performance metrics for the AutoML leader model?

U What is the current status of the H2O cloud cluster?

Checking cloud status... Your H2O cluster is 'Healthy' with 4 active nodes. Memory usage is at 35% (14GB used out of 40GB total). All hardware architecture endpoints are operational.

U Show me the last 3 training jobs

Retrieving jobs... I found 3 recent tasks: 1. GBM Training (Completed, 10m ago). 2. XGBoost Grid Search (Running, 45% complete). 3. Data Parsing (Completed, 1h ago). I can provide more details for any of these jobs.

Frequently Asked Questions

01 **How do I check memory usage with H2O.ai MCP?**

You run `cloud_status`. This tool gives you real-time diagnostics on hardware health, including how much memory is used and the total capacity of the cluster.

02 **Can I list all my machine learning models using H2O.ai MCP?**

Yes, use `list_models`. It pulls a comprehensive inventory of every model saved in your cloud instance so you know exactly what assets exist.

03 **What is the difference between list_jobs and get_model?**

`list_jobs` shows the history and current status of running tasks (like training runs). `get_model` provides the detailed metrics and configuration for a specific, already finished model asset.

04 **How do I validate data columns using H2O.ai MCP?**

First, use `list_frames` to see the available datasets. Then, specify which dataset you want and ask the agent to run `get_frame` to pull out specific column details.

05 **Does H2O.ai MCP help with data schemas?**

Absolutely. You can use `list_frames` and then `get_frame` to confirm the exact dimensional mapping and structure of your loaded datasets, ensuring schema integrity.

06 **Can my agent list all data frames currently loaded in my H2O cluster?**

Yes. Use the 'list_frames' tool. The agent retrieves the list of structured datasets securely loaded into memory, including their IDs and basic metadata, allowing you to browse available data flawlessly.

07 **How do I check the progress of a model training job via chat?**

Use the 'list_jobs' tool. Your agent will query the timeline nodes tracking all long-running tasks on the cluster, providing you with the current execution status and progress percentages synchronously.

08 **Can I see the internal architecture and metrics of a model through the agent?**

Absolutely. Use the 'get_model' tool with the specific model ID. The agent will fetch the detailed configuration blocks, exposing hyperparameters and performance metrics natively within your chat context.

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 `"h2oai": { "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

H2O.ai 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 H2O.ai. 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	H2O.ai MCP
Server ID	`019d75ad-3217-734c-8290-2b37b71ded01`
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/h2oai.