MCP SERVER | NO CODE | CLOUD HOSTED

Neptune.ai MCP

Audit model lineage from conversation.

Neptune.ai (ML Experiment Tracking) connects your agent directly to your entire machine learning lifecycle. You manage training runs, audit model versions, and inspect deep metrics without manually navigating dashboards. It gives you full, conversational control over your ML projects—from project setup to final model registry.

A+ Quality Score 100/100

mlops | experiment-tracking | model-versioning | training-metrics | data-science | telemetry

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/neptuneai-ml-experiment-tracking

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.

Neptune.ai (ML Experiment Tracking) MCP

6 tools available
Cloud-hosted on Vinkius

This MCP lets you take complete control of complex machine learning experiments using only natural conversation. Instead of clicking through multiple tabs or exporting raw CSV files just to check a metric, your agent pulls the data directly for you. You can ask it to list all active ML projects and retrieve detailed metadata configurations instantly. Need to audit performance? Your agent searches deeply across historical runs, mapping specific parameters and loss curves. It also keeps track of every model version you promote, ensuring only stable weights are available in the registry. This level of comprehensive visibility into your entire research footprint—all accessible through one unified connection via Vinkius—changes how data science works. You can verify user credentials or deep-dive into a specific project ID to get precise JSON insights on demand.

Core Capabilities

01 — View ML Project Scope

List all accessible Neptune workspaces and projects so you know the full boundaries of your work.

02 — Get Project Details

Pull specific, detailed information about a targeted machine learning project.

03 — Search Historical Runs

Find and analyze specific training runs or historical checkpoints within any given project.

04 — Inspect Model Metrics

Extract detailed telemetry, including accuracy metrics and loss curves, from a specific run's checkpoint.

05 — Manage Registered Models

List and retrieve all trained models that have been officially logged and promoted within the project.

06 — Audit User Accounts

Verify specific user credentials and confirm account availability details against your active service token.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/neptuneai-ml-experiment-tracking** — connect your AI agent in three steps.

01 Subscribe to this MCP, then enter your Neptune.ai API Token.

02 Connect the MCP to any compatible client—like Cursor or Claude.

03 Ask your agent a question, like 'Show me all registered models for Project X,' and get an immediate answer.

The bottom line is you manage complex ML lifecycles conversationally without needing to open a dedicated dashboard.

Built For

This MCP is built for technical roles that spend too much time manually cross-referencing data across multiple dashboards. If your job requires auditing model versions or comparing metrics from dozens of past experiments, you need this.

Machine Learning Engineer

Uses the agent to audit the model registry and verify experiment attributes directly from their terminal without leaving the command line.

Data Scientist

Monitors training progress and compares metrics across multiple runs by simply asking the agent, avoiding manual dashboard navigation entirely.

AI Research Lead

Tracks production model versions and ensures consistent metadata logging across several disparate ML projects efficiently.

What Changes When You Connect

01 Stop clicking through dashboards to check metrics. You ask the agent for a run's parameters, and it gives you the exact variables and loss curves instantly.

02 Keep track of version control effortlessly. Instead of guessing which model is stable, use the MCP to list and retrieve only those trained models that are marked as production-ready.

03 Simplify project visibility. You can quickly enumerate all accessible workspaces and projects, giving you a clear map of your entire ML research footprint in one go.

04 Save time auditing credentials. Need to check who has access? Use the agent to verify specific user identifiers against your service account token without manual database queries.

05 Deep-dive into data structure. Don't just get numbers; use this MCP to retrieve a precise JSON representation of any Project or Run ID for downstream processing.

Real-World Applications

Comparing the Top 3 Models

A researcher needs to compare performance across three different model architectures. Instead of running three separate reports, they ask their agent to search runs and get attributes for all three in one prompt, instantly comparing accuracy and validation loss.

Debugging a Failed Deployment

An ML engineer finds a deployed model is failing. They use the MCP to list models and then inspect specific project details, pinpointing exactly which version of the code or parameters caused the regression.

Auditing Project Scope for Compliance

A lead needs to know what ML projects exist across their department. They ask the agent to list all accessible workspaces and projects, getting a complete inventory without speaking to anyone else.

Retrieving Historical Context

An analyst needs the raw data for an old experiment run from six months ago. They use the MCP to get project details using a specific Project ID and retrieve all associated JSON metadata instantly.

Patterns to Avoid

Manual Dashboard Navigation

✗ AVOID

Opening Neptune.ai, then navigating to Projects → Select Project X → Click Runs → Filter by Date Range → Download CSV.

✓ INSTEAD

Just ask your agent: 'Show me all training runs for Project X between Q1 and Q2.' It pulls the necessary run history without any clicks.

Using Generic Search Tools

✗ AVOID

Relying on a general database tool to find metrics, which requires manually knowing the internal schema ID.

✓ INSTEAD

Use `search_runs` and then follow up with `get_attributes`. This MCP understands ML terminology, so you just ask for 'accuracy' or 'loss curve'.

Assuming Model Availability

✗ AVOID

Trying to use a model name without confirming it was promoted or logged in the current project.

✓ INSTEAD

Always run `list_models` first. This confirms that the model you want is packaged and ready for deployment within the specific project context.

The Right Fit

Use this MCP if your work involves managing the entire ML lifecycle, especially when comparing metrics across dozens of iterative runs or needing to audit which models are production-ready. If you need conversational access to metadata, run history, and model registration status, this is your tool. Don't use it if all you need is a simple database query on user records; that's better handled by a basic directory service MCP. You'll find the most value using `search_runs` combined with `get_attributes` to build a full performance picture.

The Pain of ML Experiment Auditing

Right now, checking on model performance is an exercise in context switching. You open the platform, jump between project dashboards and run histories. To compare two models, you have to manually filter runs, copy key metrics like loss curves, and paste them into a spreadsheet for comparison.

With this MCP, that entire process disappears. Your agent connects directly to your ML data source. Instead of manual clicking, you just ask: 'Compare the final accuracy of Model A vs. Model B.' You get the full comparative results instantly.

Neptune.ai (ML Experiment Tracking) for Deep Control

The biggest time sinks are figuring out which model version is safe to use and retrieving specific, deep metrics from runs that happened months ago. You spend time cross-referencing project boundaries and run IDs just to find the right JSON snippet.

This MCP gives you direct access to these checkpoints. Whether listing projects with `list_projects` or getting precise parameters via `get_attributes`, you regain full control, making model lineage traceable from a simple conversation.

Neptune.ai (ML Experiment Tracking) - 6 Tools

Use these tools to manage your ML lifecycle by listing projects, searching runs, retrieving specific metrics, and auditing model versions through natural conversation.

#	TOOL	DESCRIPTION
01	`list_projects`	Lists every Neptune workspace and project you have access to in one command.
02	`get_project`	Retrieves the specific configuration and detailed metadata for a single, named ML project.
03	`search_runs`	Searches through all tracked ML experimentation runs inside a designated project to find what you need.
04	`get_attributes`	Pulls detailed parameters and metrics logged during the runtime bounds of any specific experiment run.
05	`list_models`	Shows a list of all trained, packaged models that have been officially logged in your project.
06	`get_user`	Verifies specific user identifiers and confirms the availability status against your service account token.

See It in Action

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

U List all training runs for the 'Customer-Churn' project

I've retrieved the training history for 'Customer-Churn'. Highlights include 'churn-exp-123' (Accuracy: 0.92), 'churn-baseline' (Accuracy: 0.85), and 5 other recent runs. Would you like to see the loss curve for the top-performing run?

U Show me the metrics for run ID 'churn-exp-123'

Retrieving attributes for 'churn-exp-123'... The model achieved a final accuracy of 92.4% with a validation loss of 0.15. Learning rate was set to 0.001. No anomalous gradients were detected during the final epochs. Would you like the full JSON of all 45 logged attributes?

U List all registered models in project 'Fraud-Detection'

I've identified 3 promoted models in 'Fraud-Detection': 'XGBoost-Classifier-v2', 'RandomForest-Baseline', and 'NeuralNet-Prod-v1'. All models are mapped to production-ready weights. Which one would you like to inspect for version history?

Frequently Asked Questions

01 **How do I find a specific historical run using Neptune.ai (ML Experiment Tracking)?**

You use the `search_runs` tool to filter through all runs within a project. You can then follow up with `get_attributes` to view the deep metrics for that exact run ID.

02 **What does list_models do in Neptune.ai (ML Experiment Tracking)?**

The `list_models` tool shows you every trained model packaged and logged within your current project, confirming which weights are ready for use or promotion.

03 **Can I get all the metadata for a whole project?**

Yes. You first need to use `get_project` and provide the specific Project ID. This retrieves detailed information about its setup, boundaries, and associated resources.

04 **How do I check user permissions with Neptune.ai (ML Experiment Tracking)?**

Use the `get_user` tool. It verifies specific user credentials against your active service account token, confirming who has access to what data within the system.

05 **Is this MCP only for checking metrics?**

No. While it excels at monitoring training metrics (`get_attributes`), it also handles project visibility (`list_projects`) and model versioning (`list_models`).

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 `"neptuneai-ml-experiment-tracking": { "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

Neptune.ai (ML Experiment Tracking) 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 Neptune.ai (ML Experiment Tracking). 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	Neptune.ai (ML Experiment Tracking) MCP
Server ID	`019d75dc-6422-717d-aff0-0524e67e5167`
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/neptuneai-ml-experiment-tracking.