MCP SERVER | NO CODE | CLOUD HOSTED

# NVIDIA API Catalog MCP

## Connect your AI client to enterprise-grade compute power.

NVIDIA API Catalog MCP connects your AI client directly to a massive array of foundational models running on NVIDIA compute hardware. It lets you discover available LLMs, route complex chat queries, generate embeddings from raw text, and process visual data—all without managing individual vendor APIs.

A+ Quality Score 100/100

model-discovery | llm-proxy | inference-engine | api-catalog | model-routing | foundation-models

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

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.

# NVIDIA API Catalog MCP

8 tools available
Cloud-hosted on Vinkius

Building advanced agent workflows means connecting to dozens of specialized services. This MCP cuts through that complexity. Instead of dealing with separate credentials for every model or endpoint, your AI client talks to this central catalog. It figures out the right foundational model for the job, whether you need simple text compression or complex image analysis.

For instance, if you're building a knowledge retrieval system, your agent can first use tools like `nvidia_list_foundation_models` to see what's available. Then, it passes raw text through to `nvidia_generate_embeddings` to create vector representations. Finally, when a user asks a question, the chat completion tool handles the full conversational exchange. This centralized approach means your logic stays clean and portable. By connecting this MCP via Vinkius, you give your agent access to best-in-class GPU compute power for everything from text summarization to multimodal vision tasks.

## Core Capabilities

**01 — Discover available models**

List all explicitly hosted LLM and foundation model configurations that are currently accessible.

**02 — Route conversational chat queries**

Send unstructured text to an active LLM for immediate, contextual answers.

**03 — Generate numerical vector embeddings**

Convert raw blocks of text into dense arrays that measure semantic meaning, perfect for database searches.

**04 — Process visual data and images**

Run specialized tasks on image inputs to extract descriptions or run advanced vision analysis.

**05 — Check usage credits and limits**

Poll the system to confirm current API quota status before running expensive inference jobs.

## One Click on Vinkius — From Prompt to Execution

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

01 First, your agent sets up credentials by declaring logic tokens using the configured NVIDIA API key.

02 Next, you send a request for specific model inference, letting the MCP handle all the underlying hardware mapping and routing.

03 Finally, you receive structured completions or numerical arrays back—the data is ready to be used immediately in your application.

The bottom line is that this MCP handles the entire communication layer between your agent and massive compute resources.

## Built For

This connector is built for machine learning engineers, generative developers, and AI architects who are constantly integrating diverse models into complex systems. If you're tired of managing dozens of individual API keys just to run basic text analysis or image tagging, this MCP is what you need.

**ML Engineer**

Uses the catalog to compare different foundational models and select the best one for a specific inference task, optimizing performance.

**Generative Developer**

Builds complex workflows that chain together multiple model types—like summarizing text first, then generating embeddings, and finally using those vectors to answer questions.

**AI Architect**

Maps out the entire system architecture, ensuring that resource usage is tracked (`nvidia_check_token_quota`) across all connected model types before deployment.

## What Changes When You Connect

01 Stop worrying about model discovery. Use `nvidia_list_foundation_models` to see every available LLM path in one place, making it easy for your agent to choose the right tool for the job.

02 Handle complex resource management with `nvidia_check_token_quota`. Your workflow checks its own credit limits before running a massive inference task, preventing costly failures mid-process.

03 Need text turned into searchable data? Pass content through `nvidia_generate_embeddings` to create reliable vector arrays that power your RAG system or semantic search engine.

04 Vision tasks are now simple. Use `nvidia_vision_inference` to feed an image and get structured, actionable data back—no manual image processing needed.

05 Keep your code clean by letting the MCP handle routing. Instead of writing separate logic for summarization vs. chat, just call `nvidia_summarize_content`, and the backend takes care of the rest.

## Real-World Applications

**Building a document analysis pipeline**

A user uploads a 50-page report. The agent first uses `nvidia_list_foundation_models` to confirm capability, then passes the text to `nvidia_summarize_content`. Finally, it sends the summary and key sections through `nvidia_generate_embeddings`, allowing the end-user to search specific concepts within the document later.

**Creating a product QA bot**

The user provides an image of a complex appliance. The agent uses `nvidia_vision_inference` to extract model numbers and component names. It then passes those extracted details to `nvidia_chat_completion` to generate a tailored troubleshooting guide.

**Automating knowledge base updates**

A team uploads 100 new internal articles. The agent iterates through them, using `nvidia_generate_embeddings` on each one and storing the resulting vectors in a database. This keeps the entire knowledge base fresh for future queries.

**Testing multi-step agent logic**

Before deployment, an engineer runs a test suite that calls `nvidia_get_cloud_status` to verify latency. They then run a simulated chat session using `nvidia_chat_completion`, ensuring the entire sequence is stable and fast.

## Patterns to Avoid

### Using specific vendor APIs directly

✗ AVOID

Writing dozens of functions, each requiring unique API key management and different data structures for every single model or service you want to connect.

✓ INSTEAD

Centralize your connections. Use this MCP as a unified proxy. Your agent calls one standardized function (like `nvidia_chat_completion`), and the catalog handles the complex routing and authentication underneath.

### Ignoring resource constraints

✗ AVOID

Running an intensive, multi-step workflow that fails silently or suddenly cuts off because the API key exceeded its daily token limit.

✓ INSTEAD

Always check quotas first. Call `nvidia_check_token_quota` at the start of your job flow. This prevents failed runs and saves you time debugging usage limits.

### Handling image data manually

✗ AVOID

Writing custom code to preprocess images, resizing them, normalizing pixels, and then calling a separate vision API endpoint with complex payloads.

✓ INSTEAD

Let the tool handle it. Use `nvidia_vision_inference`. It accepts the raw input and outputs structured results directly, skipping all the manual data preparation steps.

## The Right Fit

Use this MCP if your primary challenge is *connectivity* or *complexity*. You need a single point of access to multiple specialized AI capabilities (chat, vision, embeddings) without rewriting your core agent logic every time you add a new model. This catalog pattern lets you swap out underlying models and

services seamlessly. Don't use it if all you need is a simple, one-off API call using only basic text input; in that case, a simpler, single-purpose connector might suffice. If your project requires checking system status or managing resource consumption across multiple steps, this MCP provides the necessary guardrails.

## Managing model access feels like juggling credentials.

Today, to build a single agent capable of everything—from summarizing reports to analyzing pictures—you're probably managing five or six different API keys. Every time you add a new feature, you have to check the documentation for yet another service, write custom error handling for quota issues, and map out completely separate authentication flows.

This MCP changes that. You connect once, and your agent gets access to everything. Instead of managing credentials across five different endpoints, you simply call tools like `nvidia_chat_completion` or `nvidia_vision_inference`. The system handles the routing, the keys, and the complexity for you.

## The NVIDIA API Catalog MCP delivers structured data insights.

Manual processes often leave you with raw text output that's hard to act on. You get a summary, but you can't easily search *within* the key points; or you process an image and get back a giant JSON dump that requires manual parsing.

With this MCP, if you run `nvidia_generate_embeddings`, the result is immediately useful. If you use `nvidia_summarize_content`, the output is clean and ready for the next step in your workflow. The data flows naturally from one intelligent operation to the next.

# NVIDIA API Catalog: 8 Available Tools

These tools give your agent direct access to core capabilities like running LLMs, extracting data from images, checking quotas, and listing available models.

| # | TOOL | DESCRIPTION |
|---|---|---|
| 01 | `nvidia_chat_completion` | Sends natural language questions to a hosted LLM and receives direct, generated answers. |
| 02 | `nvidia_check_token_quota` | Queries the system to check your current API usage limits and remaining credits for inference jobs. |
| 03 | `nvidia_generate_embeddings` | Takes raw text inputs and converts them into numerical vectors used for semantic search. |
| 04 | `nvidia_get_cloud_status` | Pings the core NVIDIA compute endpoints to check system latency and operational health. |
| 05 | `nvidia_list_foundation_models` | Retrieves a list of all major LLMs and foundation models that are currently available through the catalog. |
| 06 | `nvidia_list_lora_adapters` | Checks for fine-tuned model overrides, allowing you to use specialized versions without retraining the whole base model. |
| 07 | `nvidia_summarize_content` | Compresses large blocks of text into a shorter summary while retaining key information. |
| 08 | `nvidia_vision_inference` | Processes image inputs to perform advanced visual analysis and extract data from pictures. |

# See It in Action

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

**U** Deploy commands exploring active NLP data listing completely the hosted LLMs mapped heavily inside the NVIDIA catalog safely.

Parsed logically evaluating NVIDIA Cloud API natively ( `list_foundation_models` ). Platform responded safely listing 42 explicit parameters including Llama3 cleanly bounding choices naturally.

**U** Trigger inference explicitly navigating natively utilizing Nemotron LLMs to summarize standard matrices cleanly parsing bounds gracefully.

Tunnel explicitly mapping `summarize_content` . Engine successfully extracted cleanly formatted response arrays bouncing latency smoothly gracefully natively over hosted limits.

**U** Execute explicitly generating explicit unstructured text matrices extracting native embedding queries purely isolating the arrays properly.

Execution logic parameters strictly extracting values safely allocating implicitly natively `generate_embeddings` . Payload correctly returned arrays naturally formatting vector bindings efficiently bounds.

# Frequently Asked Questions

01 **How do I check if a model exists before calling nvidia_chat_completion?**

You should run `nvidia_list_foundation_models` first. This tool dumps an array of all accessible LLM paths, letting you confirm the exact model name your agent needs to use.

02 **Does this MCP handle API quota issues?**

Yes. You can proactively run `nvidia_check_token_quota` at the beginning of any workflow. This tells your agent exactly how many credits are left, stopping runs before they fail due to overage.

03 **What is the difference between nvidia_generate_embeddings and chat completion?**

Chat completion generates conversational text responses. Generating embeddings converts unstructured text into dense numerical arrays, which you use for semantic search or clustering, not conversation.

04 **Can I process images with this MCP?**

Yes. Use the `nvidia_vision_inference` tool. It specifically handles multimodal tasks, allowing your agent to run advanced analysis on visual data.

# 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 `"nvidia-api-catalog": { "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

# NVIDIA API Catalog 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 NVIDIA API Catalog. 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 | NVIDIA API Catalog MCP |
| Server ID | `019d75e1-35ae-70cf-91e7-31316ddc2c23` |
| 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/nvidia-api-catalog.