MCP SERVER | NO CODE | CLOUD HOSTED

Replicate MCP

Run and manage thousands of ML model predictions.

Replicate lets your AI agent access thousands of open-source machine learning models—for generating images, text, audio, and video. Instead of jumping between web dashboards or writing complex API calls, you talk to your agent, and it handles the entire ML lifecycle: finding a model, setting parameters, running the prediction, and retrieving the final result.

A+ Quality Score 98.33/100

machine-learning | model-inference | generative-ai | api-integration | cloud-computing

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/replicate-alternative

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.

Replicate MCP

12 tools available
Cloud-hosted on Vinkius

Your AI client connects directly to this MCP to treat open-source ML models like an internal service. You can ask your agent to find specific capabilities—like text-to-image generators or advanced LLMs—and it handles model discovery and selection across thousands of available options. Need to run something? Just tell your agent what you want, and it executes the prediction. It tracks everything from 'starting' to 'succeeded', giving you a single conversation thread for complex ML operations. The whole process is abstracted away; you don't manage API keys or wait on status pages. All this power is housed within Vinkius, making Replicate an operational resource available through any MCP-compatible client.

Core Capabilities

01 — Discovering Model Capabilities

Your agent finds and details specific ML models by name or category.

02 — Finding Related Models

You can list entire groups of related models, such as all text-to-image generators or all LLMs.

03 — Checking Account Status

The agent verifies your token status and shows you current usage information.

04 — Launching Predictions

You initiate a model run by providing the necessary input data to generate content.

05 — Tracking Results

The agent monitors running predictions, telling you when they start, process, fail, or finish.

06 — Managing Resources

You can view available GPU hardware options and list your prediction history.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Replicate API Token.

02 Tell your AI agent what you need, like 'Generate a picture of a robot reading' or 'List all video models'.

03 The agent runs the necessary tool calls in the background, providing you with the status updates and final output links directly in your conversation.

The bottom line is you treat complex machine learning pipelines like simple conversational commands.

Built For

The ML Engineer who needs to prototype dozens of models without touching the command line. The Developer who wants to build an AI application that generates content, not just text. The Researcher who needs to compare hardware requirements for multiple model types.

ML Engineer

They use this MCP to compare different models by checking their required GPU hardware before deciding which one is cost-effective for a new project.

Software Developer

They integrate prediction status tracking into an application, using the agent to check if a generated image or audio file is ready without constant polling.

AI Researcher

They use this MCP to quickly explore model collections and inspect version schemas for models they plan to test in a new environment.

What Changes When You Connect

01 Stop managing multiple websites. Instead of navigating the Replicate site to check status, you simply ask your agent for the prediction status using `get_prediction` or review history with `list_predictions`. It's all in one conversation.

02 You don't need to guess what models exist. Use `search_models` or `list_collections` to quickly discover everything available—from text-to-image generators to video processors—without leaving your chat window.

03 Model setup used to be a pain, requiring you to find the right version ID. Now, use `get_model_versions` to inspect the full schema and get the correct ID before running a prediction with `create_prediction`.

04 Managing costs is easier when you can check hardware options. Use `list_hardware` to see available GPU types and pricing tiers before launching any job, preventing expensive mistakes.

05 The ability to cancel jobs mid-stream is huge. If you realize the prompt was wrong after a few seconds, use `cancel_prediction` immediately instead of letting it run to completion.

Real-World Applications

Creating an AI art campaign

A marketer needs 50 different fantasy images for a product launch. Instead of manually running fifty separate commands, they ask their agent to search for the best text-to-image model, run five variations, and track all the outputs using `create_prediction` and `get_prediction`.

Testing LLM performance

A developer needs to compare how three different Large Language Models (LLMs) handle a specific set of prompts. They use `search_models` to find the best candidates, then run multiple predictions, and finally review their usage logs using `list_predictions`.

Building an automated video pipeline

A content creator wants to turn a text description into a short video. They first check available hardware with `list_hardware`, select the right model, and run the prediction, ensuring they get all necessary status updates via `get_prediction`.

Debugging an ML pipeline

An ML engineer runs a batch of predictions but one fails. Instead of checking logs manually, they use `list_predictions` to see the failure ID and then check the details using `get_prediction` to understand why it failed.

Patterns to Avoid

Guessing model parameters

✗ AVOID

The user tries to run a prediction for an LLM but doesn't know if they need the `owner/name` format or what input schema is required, leading to immediate failure.

✓ INSTEAD

First, use `get_model` with the specific owner and name slug. This confirms the model exists and shows you its exact input requirements before attempting any prediction.

Ignoring usage costs

✗ AVOID

A developer runs multiple image generation jobs back-to-back without knowing if they are using a powerful, expensive GPU cluster, resulting in unexpected billing.

✓ INSTEAD

Always check available hardware and pricing first by running `list_hardware`. This gives you the cost context needed to plan your workload before calling `create_prediction`.

Treating ML like simple APIs

✗ AVOID

The user expects a single command to instantly return all results. Since models take time (10-60 seconds), the prediction will fail if not checked later.

✓ INSTEAD

After `create_prediction`, you must use `get_prediction` periodically. This tool is designed specifically for checking status and retrieving final results once they are ready.

The Right Fit

Use this MCP if your primary bottleneck is the operational complexity of running ML models, not the creativity itself. Specifically, if you need to discover thousands of tools (models), manage their lifecycle (creation, tracking, cancellation), or compare resource needs (hardware/versions) before execution. You'll use it if your workflow involves a sequence: Search -> Select Model/Version -> Run Prediction -> Check Status.

Don't use this MCP if you just need to list model names; `list_models` handles that simply. Also, don't use it if you are only trying to understand the general concept of generative AI—that requires reading documentation, not running code. If your goal is purely resource management (like billing reports), look for dedicated accounting services instead.

The friction points in ML prototyping today are brutal.

Right now, generating content with open-source AI models feels like a multi-tab web session. You check the Replicate website for available models; you copy complex slugs and schemas into your local script; then, to see if it worked, you have to wait until the prediction finishes, manually checking its status on another page.

With this MCP, all that complexity vanishes. Your agent handles the entire process conversationally: finding the model, submitting the data, and giving you a clear update when the content is ready. You just get the result.

Replicate MCP gives you full ML lifecycle control.

You no longer have to manually switch between discovery, execution, and status checking. Tools like `get_model` give you deep model details upfront; `create_prediction` runs the job; and `get_prediction` tracks the outcome—all within one conversation.

What's different now is that your AI client acts as a dedicated ML operations assistant, keeping track of everything so you can focus purely on what you want to create.

Replicate Alternative: 12 Tools for Model Operations

These twelve tools allow your AI client to manage the entire machine learning lifecycle, from discovering models to running predictions and tracking final results.

#	TOOL	DESCRIPTION
01	`cancel_prediction`	Stops an ML prediction that is currently running using its unique ID.
02	`create_prediction`	Starts a new model run by sending the required inputs and specifying the target model.
03	`get_account`	Checks your API token status, showing your account type and usage limits.
04	`get_collection`	Retrieves details for a specific group of models, like all audio-to-audio effects.
05	`get_model`	Fetches detailed information about a single ML model using its full owner/name slug.
06	`get_model_versions`	Lists all available versions for a specific model, including their IDs and required schemas.
07	`get_prediction`	Retrieves the current status and final output data of any prediction using its ID.
08	`list_collections`	Shows all available model collections, grouping models by type (e.g., text-to-image).
09	`list_hardware`	Provides a list of available GPU hardware options and their pricing tiers.
10	`list_models`	Displays a broad catalog of every model, including run counts and required hardware.
11	`list_predictions`	Shows the history of your most recent runs, giving IDs, models, and status for tracking.
12	`search_models`	Narrows down the catalog to find specific types of models using a keyword query (e.g., 'music' or 'llm').

See It in Action

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

U List all text-to-image collections on Replicate.

Found the 'text-to-image' collection with featured models including stability-ai/sdxl, black-forest-labs/flux-schnell, and ideogram-ai/ideogram-v2. The collection has 50+ models total.

U Search for LLM models on Replicate.

Found popular LLM models: meta/meta-llama-3-70b-instruct (2M+ runs), mistralai/mistral-7b-instruct-v0.3 (1.5M+ runs), google/gemma-2-27b-it (800K+ runs). Each model shows hardware requirements and example inputs.

U Create a prediction using stability-ai/sdxl with prompt 'a sunset over mountains, photorealistic'.

Created prediction pred_abc123. Status: starting. Check back with `get_prediction` to retrieve the generated image URL once it completes (usually 10-30 seconds).

Frequently Asked Questions

01 **How do I find out what models are available in Replicate using the Replicate MCP?**

Use `list_models` to get a broad overview of every model. For more focused results, try `search_models`, which lets you narrow down by keywords like 'llm' or 'video'.

02 **What if my prediction fails? How do I check the error details with Replicate MCP?**

Use `get_prediction` and provide the failed ID. This tool returns logs and status information, helping you diagnose whether the failure was due to bad input or a model issue.

03 **Does the Replicate MCP help me manage costs?**

Yes. Before running any job, check available options using `list_hardware` to see GPU types and associated pricing for your prediction workload.

04 **Can I run a model if I don't know the exact version ID? (Replicate MCP)**

No. To ensure compatibility, you must first use `get_model_versions` to find all versions of the model and select the correct 64-character hash ID for `create_prediction`.

05 **What is the difference between `list_models` and `search_models` on the Replicate MCP?**

`list_models` gives you a full directory of everything. `search_models` lets you filter that massive catalog based on specific keywords, making discovery much faster.

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 `"replicate-alternative": { "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

Replicate 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 Replicate. 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	Replicate MCP
Server ID	`019d8477-8851-70ce-8501-78d3fa84df45`
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/replicate-alternative.