MCP SERVER | NO CODE | CLOUD HOSTED

Replicate MCP

Run open-source ML workflows from chat.

Replicate MCP lets your AI client dynamically search, run, and manage thousands of open-source machine learning models. You can command complex tasks—like generating images, running specialized language models, or processing audio—directly from a chat prompt using natural language instructions.

A+ Quality Score 100/100

machine-learning | model-inference | open-source-models | fine-tuning | api-access | generative-ai

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

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

This connector gives your agent the power to interact with a massive library of open-source ML models without needing to run them on your own hardware. Instead of dealing with complex API calls and parameter files, you simply tell your AI client what you want done in plain English. It handles finding the right model, checking its required inputs, starting the job, and even monitoring it until it's finished.

Need a specific type of image? Your agent can search for models and then execute a prediction with just a few words. If the process is long-running, you don't have to wait by the console; your AI client manages the status updates automatically. It’s a huge step up from traditional methods. When you connect this capability through Vinkius, you get instant access to the entire catalog of model operations, making complex ML workflows manageable right inside your chat interface.

Core Capabilities

01 — Find and list available models

It lets your AI client search across thousands of public model definitions based on a keyword or use case.

02 — Execute ML predictions

You can start running specific open-source models, providing the necessary input variables to generate output like images or text.

03 — Manage job status and lifecycle

Your AI client tracks ongoing jobs, retrieving the results when they're ready or canceling them immediately if you change your mind.

One Click on Vinkius — From Prompt to Execution

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

01 First, install the Replicate platform extension module into your MCP.

02 Next, input your personal Replicate API Token into the configuration variables.

03 Finally, prompt your agent naturally: 'Search for a video generation model, check its parameters, and generate a clip of a cat on Mars.'

The bottom line is that you tell your AI client what task to complete, and it handles all the necessary backend steps.

Built For

This MCP is for developers, content creators, and data scientists who are tired of manually logging into ML platforms or writing boiler-plate code just to test a model. If you need to quickly prototype with diverse open-source models, this connector saves hours.

AI Developer

You use it to rapidly prototype and integrate novel algorithms by running quick predictions without modifying local Python notebooks.

Content Creator

You delegate specialized tasks —like generating unique audio or high-quality visuals—directly from your chat interface instead of using multiple external web tools.

Machine Learning Researcher

You test and compare the output of many different open-source models quickly, systematically checking model metadata to ensure parameter requirements are met before running a prediction.

What Changes When You Connect

01 Access diverse models instantly. You don't need to hardcode API endpoints; just tell your agent what kind of image or text you want, and it handles the search using `search_models`.

02 Manage long jobs without stress. If a video generation task takes minutes, use `get_prediction` to check its status later or call `cancel_prediction` if the results aren't right.

03 Stop guessing parameters. Before running anything, use `get_model` to pull up the exact schema and input requirements for any model you find, preventing failed runs.

04 Run models without local setup. This MCP lets your agent connect directly to powerful cloud infrastructure, bypassing the need to install Python dependencies or manage GPU drivers locally.

05 Build complex chains easily. You can instruct your AI client to take the output of one specialized model and feed it as input to a second model using natural language instructions.

Real-World Applications

Generating marketing assets for a new product launch.

A content manager needs 20 unique concept images. Instead of writing a script that iterates through image generation APIs, they prompt their agent: 'Find five text-to-image models and generate ten variations for this car design.' The agent uses `search_models` to find options, then executes multiple predictions.

Analyzing user feedback audio files.

A researcher wants to test different speech-to-text or text-to-speech models. They use their agent to execute a prediction on an audio file, and if the results are poor, they can immediately call `list_predictions` to check historical logs for better model versions.

Prototyping an LLM feature for a client.

A developer wants to test how different language models handle specific JSON inputs. They use the agent's ability to `get_model` metadata first, ensuring they provide the correct payload structure before calling `create_prediction`.

Monitoring a large batch of scientific simulations.

A scientist kicks off 50 complex climate models. Instead of checking every dashboard, they ask their agent to monitor all jobs using `list_predictions`, getting real-time status updates until the final output is retrieved via `get_prediction`.

Patterns to Avoid

Assuming a model works with basic prompts

✗ AVOID

The user types: 'Generate an image of Mars.' and the prediction fails because they didn't specify required parameters like aspect ratio or seed.

✓ INSTEAD

First, use `search_models` to find relevant tools. Then, before running it, call `get_model` on that specific model ID. This step exposes the exact JSON structure needed for a successful run.

Ignoring job status

✗ AVOID

The user runs a long process and then forgets about it, assuming the result is instantly available or failed silently.

✓ INSTEAD

Always confirm the job status. Use `get_prediction` to reliably check if your running task has finished its cycle before attempting to access the output data.

Overloading the agent with too many actions at once

✗ AVOID

The user tries to list collections, search models, and run a prediction all in one single prompt, confusing the agent's intent.

✓ INSTEAD

Break it down. Use `list_collections` first to narrow your options, then use `search_models` with a specific keyword from that collection, and *finally* call `get_model` for the precise setup.

The Right Fit

Use this MCP if your core problem is model orchestration: you need an AI agent to discover, configure, run, and monitor diverse open-source machine learning models hosted on Replicate. You're working with varied inputs (audio, images, text) and the output requires a multi-step process of discovery followed by execution.

Don't use this if: 1) Your need is simply to call one specific API endpoint repeatedly without variation; an SDK might be cleaner. 2) You are only dealing with structured data (like reading from a database); a dedicated data connector is better. This MCP excels when the *process* of finding and running the model is as important as the result itself.

The tedious cycle of ML prototyping today

You know the drill: you need to test a new image generation model. You open the documentation, copy-paste the required JSON payload structure into your local script, run it, see an error because you missed one mandatory variable, then have to manually check the platform logs to figure out what went wrong. It's a constant cycle of copying parameters and managing different dashboards.

With this MCP, that process vanishes. You tell your agent in plain English: 'Find me a good model for sci-fi concepts.' The agent handles the search (`search_models`), checks the requirements (`get_model`), and then runs the job (`create_prediction`). You get results—not error logs.

Replicate MCP gives you instant, delegated ML power

You no longer have to write wrapper code for every single model or manually check which parameters are required. The agent does the discovery work for you—finding collections (`list_collections`) and checking deployments (`list_deployments`) automatically.

Your workflow shifts from 'How do I call this API?' to 'What do I want to create?' It's about delegating complex, multi-step technical tasks to your AI client. Period.

Replicate MCP with 12 Tools

Use these tools to search for models, manage deployments, track job status, and execute complex machine learning predictions.

#	TOOL	DESCRIPTION
01	list_models	It shows you a list of all public machine learning models available on Replicate.
02	get_account	This retrieves basic information about your connected Replicate account details.
03	list_collections	It lists curated groups of models, such as those focused on 'Image-to-Text' or 'Audio Generation'.
04	list_deployments	This shows you all the active model deployments you have set up personally.
05	cancel_prediction	It stops a model prediction job that is currently running and prevents further processing.
06	create_prediction	You start a new model prediction by supplying the required model version ID and all necessary inputs as a JSON object.
07	get_collection	It retrieves details for a specific, defined group of models using its unique slug.
08	get_model	This fetches detailed information about one specific model, including its exact owner and parameter requirements.
09	get_prediction	It checks the current status of a prediction job and retrieves the final output if it has finished running.
10	list_hardware	This lists all available GPU hardware options you can use for running your models.
11	list_predictions	It retrieves a log of the recent prediction jobs that have been run by your account.
12	search_models	You can search across the entire platform to find public models that match specific keywords or use cases.

See It in Action

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

U List my recent predictions.

Invoking `list_predictions` has successfully found your last requests. The most recent executed instance has an ID of `p_30abc...`, which is confirmed finished, running a prompt about dog pictures.

U Query Replicate to search for 'TTS' models.

I ran `search_models` using your keyword "TTS". Replicate returned a wide array of options, including 'suno-ai/bark' and 'coqui/xtts'. Please specify the precise owner/model so I can inspect their schematics thoroughly!

U Cancel the prediction that has the ID `p_abc123`.

I immediately routed the termination request using the tool `cancel_prediction` targeted at your specified iteration ID `p_abc123`. The model sequence is permanently halted, halting processing instantly.

Frequently Asked Questions

01 **Can the Replicate MCP handle image generation?**

Yes, absolutely. You can command your agent to find and run specific text-to-image models by calling `search_models` and then executing a prediction.

02 **What is the difference between `list_collections` and `search_models` in Replicate MCP?**

`List_collections` shows pre-curated groups of related models (like all 'Audio Generation' tools). `Search_models` lets you search across every single model on the platform using keywords.

03 **How do I stop a job running with Replicate MCP?**

If a prediction is taking too long or isn't giving the right result, use `cancel_prediction` to halt it immediately and cleanly. This prevents unnecessary usage costs.

04 **Does Replicate MCP require me to run models on my own computer?**

No. The entire purpose of this MCP is that your agent connects to the cloud infrastructure, so you never have to worry about local hardware or setup conflicts.

05 **What if I want to see a history of my past model runs using Replicate MCP?**

You can check your recent activity by calling `list_predictions`. This tool gives you an immediate log of all the jobs that have been run through this MCP.

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": { "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	`019d75fe-9426-7272-9964-c32556c42621`
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.