MCP SERVER | NO CODE | CLOUD HOSTED

Playground AI MCP

Generate or modify high-res visuals with natural language.

Playground AI lets your agent generate, modify, and upscale high-resolution images using only natural language instructions. Stop opening separate art programs; instead, tell your LLM to create assets—whether you need a simple placeholder or complex architectural rendering—and get the result instantly within your workflow. It handles everything from background removal to expanding image borders with precision.

A+ Quality Score 100/100

generative-art | image-generation | inpainting | outpainting | ai-models | creative-workflow

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

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.

Playground AI MCP

10 tools available
Cloud-hosted on Vinkius

This MCP connects your AI client directly to Playground AI's powerful compute clusters. Instead of manually dragging images into an external editor, you just instruct your agent what visual asset you need. You can ask it to generate entirely new graphics based on a text prompt, or take an existing sketch and refine it using specific guidance types like depth maps. If you need to fix a photo by filling in missing sections, the MCP handles that surgically. It's also perfect for scaling small icons up to 4x resolution without losing detail. When your AI agent needs this kind of visual power, connecting through Vinkius lets you access it from any compatible client, keeping your entire creative pipeline inside your existing code environment.

Core Capabilities

01 — Generate images from text prompts

Create new assets instantly by giving a simple description and specifying size constraints.

02 — Adjust existing visuals with guidance

Change an image's style or content using a prompt while controlling how much the original image should be retained.

03 — Fill in missing parts of an image

Use a mask to tell the AI exactly which area needs content, generating realistic fills over defined regions.

04 — Enlarge and expand images

Scale blurry pictures up to 4x for high detail, or instruct the model to generate new background context outside the original frame.

05 — Isolate subjects from backgrounds

Automatically remove a subject's surroundings and return a clean asset with a transparent background.

One Click on Vinkius — From Prompt to Execution

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

01 Append the Playground AI integration to your MCP setup and provide your API key.

02 Instruct your agent using plain language (e.g., 'Generate a 1024×1024 image of...').

03 The system executes the request, and you receive the final rendered asset URL back in your code output.

The bottom line is that you treat complex visual editing like writing another function call; it's just a prompt away.

Built For

Anyone who spends time iterating on visual assets—web developers building mockups, concept artists needing rapid background removal, or creative directors constantly refining marketing visuals. You're tired of jumping between Photoshop and your IDE just to make one change.

Web Developer

Needs perfectly sized graphical placeholders or hero images generated without opening a dedicated visual editor.

Concept Artist

Requires automating repetitive tasks like background removal and quick upscaling of preliminary sketches to pitch quality.

Creative Director

Needs rapid, iterative visual adjustments—like expanding a scene or changing the mood—while discussing copy in a chat interface.

What Changes When You Connect

01 It eliminates the need to manually open a separate art application. You simply instruct your agent, and it executes complex visual tasks like `generate_image` right in your workflow.

02 Achieve professional isolation of subjects instantly using the `remove_background` tool. This gives you perfectly clean assets without having to edit them in another program.

03 You can dramatically expand a photo's context using `outpaint_image`. If your scene is too narrow, tell the agent to generate the missing environment around the edges.

04 For maximum control, use `generate_with_controlnet`. This lets you guide the AI not just with text, but with structural inputs like depth or pose maps for reliable results.

05 Upscaling blurry or low-res assets is simple. Running `upscale_image` boosts quality by 2x or 4x, ensuring your final graphic looks crisp everywhere.

Real-World Applications

Building a Product Mockup Gallery

A web developer needs to populate a gallery of placeholder product images. Instead of manually generating multiple assets, they prompt their agent: 'Generate six 1024×1024 images using `generate_image` depicting different views of the widget.' The MCP returns all necessary assets immediately.

Marketing Campaign Asset Refresh

A creative director has a product photo but needs to change the environment. They use `outpaint_image` on the original picture, prompting the AI to extend the background from an indoor setting into a sunny outdoor market scene.

Cleaning Up Source Material

A concept artist receives a messy sketch and needs to clean up the foreground character. They use `remove_background` first, then `inpaint_image` with an explicit mask over clothing details, ensuring structural consistency.

Improving Low-Quality Inputs

A team member has a low-resolution logo graphic that looks pixelated on the website. They pass the image URL to `upscale_image`, boosting it 4x and making it suitable for high-DPI screens.

Patterns to Avoid

Trying to use multiple tools manually

✗ AVOID

The user generates an image, then has to copy the URL into a separate upscaling tool, and finally paste it into an inpainting mask. This is slow and error-prone.

✓ INSTEAD

Chain the commands within your agent: first, generate the base image using `generate_image`. Then, immediately call `upscale_image` on that result's URL, followed by `inpaint_image` if you need to fix a specific area.

Assuming simple text prompts are enough

✗ AVOID

The user asks the agent for an image of 'a cat in space.' The output is generic and lacks the desired architectural style.

✓ INSTEAD

Refine your input by specifying constraints. Use `generate_with_controlnet` and provide a reference image that captures the specific architecture or pose you need to guide the AI.

Forgetting about dimensions

✗ AVOID

Generating an asset without defining the required width and height, leading to unusable aspect ratios for web components.

✓ INSTEAD

Always include spatial constraints when generating. Call `generate_image` and explicitly define the needed dimensions (e.g., 1024×1024) in your prompt.

The Right Fit

Use this MCP if your primary goal is visual iteration or asset creation that requires multiple distinct steps, like 'Generate -> Upscale -> Mask -> Refine.' It's built for complex workflows where the output of one step becomes the input for the next. Don't use it if you just need a simple text-to-text summary; those tools are better suited for pure data analysis. If your task is simply to fetch metadata about existing assets, using `list_generations` or `get_generation` is sufficient. However, if you need to *change* the pixels themselves—whether by expanding space (`outpaint_image`) or cleaning up borders (`remove_background`)—this MCP provides the necessary full suite of visual controls.

The Friction of Visual Workflows

Right now, making a single high-quality asset is a multi-step process. You start in your IDE, get an idea, then you have to copy the prompt into Playground's website. Next, if you need to crop it or remove a background element, you open a second tool, manually upload the file, wait for it to process, download the result, and finally paste it back into your code mockup. It’s clicking through three or four different platforms just to get one usable image.

With this MCP, that entire sequence collapses into a single instruction set for your agent. You tell your client: 'Generate an asset, then expand its borders, then remove the background.' The system handles all those intermediate uploads and processing steps automatically. What you get is not just an image; it's a finished, fully processed asset ready to drop straight into your application.

Playground AI MCP: Generating Complex Visual Assets

The manual steps that disappear are the context switching and the intermediate file handling. No more copying URLs from one tab to another. Your agent now controls the entire pipeline, managing the inputs for `inpaint_image` masks, passing results through `upscale_image`, and using those outputs immediately in a subsequent generation step.

This means your creative process isn't bottlenecked by UI limitations or context switching. You talk to your AI client like you’re talking to a specialized art department assistant who can execute any visual task instantly, every time.

Playground AI: 10 Image Generation Tools

These tools allow your AI client to perform every major visual task—from initial generation to complex restoration—using a single, unified interface.

#	TOOL	DESCRIPTION
01	`generate_image`	Creates new images from scratch using a text prompt, allowing you to set specific dimensions and select multiple models.
02	`transform_image`	Alters an existing image's style or content based on a prompt, controlling the degree of change via a strength setting.
03	`inpaint_image`	Fills in specific missing areas within an image using a provided mask and descriptive text.
04	`upscale_image`	Increases the resolution of an existing picture, enhancing both detail and overall clarity by a factor of 2 or 4.
05	`remove_background`	Separates a subject from its surroundings and returns a clean asset with a transparent background.
06	`outpaint_image`	Extends an image beyond its current borders, generating new content that matches the specified direction (up, down, left, right).
07	`generate_with_controlnet`	Creates images while forcing structural guidance using reference types like depth maps or edge detection.
08	`list_models`	Returns a list of all available AI models, detailing their capabilities and names for use in your prompts.
09	`get_generation`	Retrieves the full details, prompt, and metadata for a specific image generation ID.
10	`list_generations`	Lists records of your recent visual asset creations, including IDs and timestamps.

See It in Action

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

U Generate a 1024×1024 image of a cyberpunk coffee cup in neon lighting.

Payload dispatched holding spatial constraints to 1024×1024 pixels. Using default Model 'Playground v3'. The system returns generation block `ID: 9x8A-CUP`. The finalized render URL is ready: [View Graphic]. Want to apply an iterative 2x Upscale on it now?

U Upscale this image to 4x its size `https://example.com/small_icon.png`.

Upscaling protocol engaged. Submitted target to `upscale_image` invoking factor constraint '4'. The engine interpolated logical pixels correctly matching standard textures natively. Result pointer: [High-Res URL].

U Remove the background from the image at `https://example.com/person.jpg`.

Background segmentation dispatched via `remove_background`. The deep learning layer masked the subject heavily and omitted out-of-bounds layer fragments. I've received the pristine transparent PNG rendering back. The new asset is currently available at `[Extracted PNG URL]`. Would you like me to map it in a new CSS class?

Frequently Asked Questions

01 **How do I use Playground AI MCP for image generation?**

You initiate creation by using the `generate_image` tool. You simply pass a descriptive text prompt and specify the required dimensions (width and height) to get started.

02 **Can I enlarge a small image with Playground AI MCP?**

Yes, use the `upscale_image` tool. You can pass an existing image URL and specify whether you want a 2x or 4x increase in resolution.

03 **What is the best way to change an image's style using Playground AI MCP?**

Use `transform_image`. You provide your public image URL, a text prompt describing the new look, and a strength value (0-1) to control how much the original photo changes.

04 **Does Playground AI MCP support background removal?**

Yes, the `remove_background` tool automatically segments the subject from its surroundings, giving you a high-quality, transparent PNG asset.

05 **How do I expand an image's scene using Playground AI MCP?**

Use `outpaint_image`. You pass your original picture URL and specify the direction (up/down/left/right) to instruct the model on generating new 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 `"playground-ai": { "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

Playground 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 Playground 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	Playground AI MCP
Server ID	`019d75f7-40ba-7030-9962-d75330b41868`
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/playground-ai.

© 2026 Vinkius · vinkius.com · support@vinkius.com