MCP SERVER | NO CODE | CLOUD HOSTED

Helius (Solana) MCP

Analyze live Solana balances and full transaction history.

Helius (Solana) gives your AI agent real-time access to Solana blockchain data. It lets you pull asset metadata, check wallet balances with live USD pricing, track complete transaction history, and get precise network fee estimates directly through natural language queries.

F Quality Score 11.43/100

solana | rpc | blockchain-data | nft | web3 | transaction-history

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

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Helius (Solana) MCP

14 tools available
Cloud-hosted on Vinkius

Need deep insights into the Solana blockchain without writing complex API calls? This MCP connects your AI client to Helius's robust data infrastructure. You can ask it anything about a wallet or an asset—from listing every owned token to calculating how much gas you'll need for a transaction.

It fetches detailed metadata for everything from fungible tokens to compressed NFTs, giving you full visibility into ownership and history. If your job involves crypto forensics or tracking complex digital assets, this is it. It lets you see a wallet's complete balance sheet, including the USD value of all held assets, and track every incoming and outgoing transfer. Because Vinkius hosts this MCP in its catalog, you can connect it once from any AI client—be it your IDE or desktop app—and start querying Solana data immediately.

Core Capabilities

01 — List a wallet's entire asset portfolio

Find every NFT and token owned by a specific address.

02 — Determine live financial balances

Get the current holdings of an address, including real-time USD pricing for all assets.

03 — Track full transaction movements

Retrieve a complete log of all incoming and outgoing transfers for any wallet.

04 — Forecast network gas costs

Calculate the recommended priority fee needed to ensure your transaction processes quickly, even during peak congestion.

05 — Search assets by traits or collection

Query the blockchain for specific digital items using advanced filters like creator or attribute.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your Helius API Key within Vinkius.

02 Instruct your AI client (your agent) to analyze a specific Solana wallet address or asset ID.

03 The system runs the query against the blockchain data, returning structured results like balances, transaction logs, or metadata.

The bottom line is you get clean, actionable data about any Solana asset or wallet without leaving your preferred development environment.

Built For

This MCP is essential for crypto analysts and web3 developers who spend their days verifying on-chain state. If you're tired of cross-referencing multiple dashboards to build a complete asset picture, this tool gives your agent the full visibility it needs.

Crypto Analyst

You use it to perform deep wallet forensics, checking transaction history and calculating total portfolio value by calling get_wallet_history or get_wallet_balances.

Web3 Developer

You connect this MCP to your IDE to debug smart contracts, verifying on-chain states and asset ownership using get_assets_by_owner before testing code.

NFT Product Manager

You use it to audit collections or track specific items' metadata by running advanced searches via search_assets or getting detailed item data with get_asset.

What Changes When You Connect

01 Stop guessing about gas fees. Use get_priority_fee_estimate to calculate the exact network fee needed, ensuring your critical transactions process fast even when the chain is slammed with activity.

02 Get a true picture of value instantly. The get_wallet_balances tool returns all token and NFT holdings alongside their current USD price, giving you an immediate, accurate portfolio valuation.

03 Audit ownership effortlessly. Running get_assets_by_owner quickly lists every digital item owned by any address, letting analysts build comprehensive asset reports in seconds.

04 Understand the flow of money better. You can use get_wallet_transfers and get_transaction_history together to map out exactly where funds are moving in or out of an account over time.

05 Search without limits. When you need to find a specific type of digital collectible, search_assets lets you filter assets using advanced criteria like creator or collection name.

Real-World Applications

Investigating suspicious funding sources

A compliance officer needs to know where a newly funded wallet got its money. Instead of manually checking multiple records, they ask their agent to use get_wallet_funded_by and then check the transaction log using get_transaction_history. The agent immediately reports the original source.

Auditing an NFT collection for ownership gaps

A PM needs to verify which owners hold a specific metadata trait across 10,000 assets. They instruct their AI client to use search_assets with advanced filtering. The agent returns a filtered list of addresses and the required asset details.

Debugging contract deployment failure

A developer runs into an unknown transaction error. Instead of spending hours digging through raw data, they run parse_transactions on the failed payload. The agent immediately translates the cryptic codes into plain English, pointing out the specific issue.

Tracking a complex cross-platform asset movement

A collector wants to know every item an address has ever owned. They ask their agent to use get_assets_by_owner, followed by get_token_accounts to confirm the underlying mints and total balances.

Patterns to Avoid

Confusing general search with asset retrieval

✗ AVOID

The user thinks they can just ask for 'all assets' without specifying filters, getting a massive, unusable dump of data that includes irrelevant records.

✓ INSTEAD

To get precise results, use advanced filtering tools. Start by using search_assets if you know the collection or creator, or run get_assets_by_owner to list everything held by an address.

Ignoring balance valuation

✗ AVOID

The user pulls raw token data but has no idea what it's worth in real dollars, leading to poor financial reporting.

✓ INSTEAD

Always run get_wallet_balances. This tool combines your total holdings with live USD pricing, giving you an accurate, single-number valuation immediately.

Overlooking transaction detail

✗ AVOID

The user only checks the basic history and misses key details like whether the funds came from a specific source or if the transfer was partial.

✓ INSTEAD

To get full context, use get_wallet_history to see everything. If you need more granularity on incoming/outgoing flows, run get_wallet_transfers.

The Right Fit

Use this MCP if your workflow requires deep, real-time data access to the Solana blockchain ledger. Specifically, if you need to know *what* an address owns (get_assets_by_owner), *how much* it's worth right now (get_wallet_balances), or *where* its funds originated (get_wallet_funded_by). You should also use it when debugging complex on-chain interactions, as tools like get_asset and parse_transactions help interpret raw data. Don't use this if you just need basic market price feeds; that requires a simple API endpoint,

not full blockchain query capability. If your goal is only to track Ethereum assets or Polygon holdings, you need an entirely different MCP for those networks.

The struggle of stitching together asset and wallet data manually.

Today, figuring out the true value or history of a crypto address means jumping between multiple dashboards. You check one tab for balances, another for transfers, and a third for specific NFT metadata. Then you copy-paste everything into a spreadsheet just to get a coherent picture. It's tedious, time-consuming, and easy to miss crucial data points.

With this MCP, your agent handles the heavy lifting. You simply ask, 'What is the total value of assets held by address X?' The system automatically calls multiple tools—getting balances, fetching metadata, and calculating current USD values—and delivers one clean answer. Your AI client does the multi-step data integration for you.

Helius (Solana) MCP provides full asset visibility.

You no longer need to manually query every single type of asset or track transfers across multiple endpoints. Instead, the agent can list all assets owned by an address using get_assets_by_owner and then instantly pull detailed metadata for any specific item using get_asset.

The result is a complete audit trail delivered in plain text. You stop wasting time clicking through tabs and start getting answers.

Helius (Solana) MCP: 14 Tools Available

These tools allow you to query everything from asset ownership and token balances to network fee estimates directly from the Solana blockchain.

#	TOOL	DESCRIPTION
01	`get_asset_batch`	Retrieves the detailed information for multiple assets in a single API call.
02	`get_asset_proof`	Generates a cryptographic proof (Merkle proof) specifically for compressed NFTs.
03	`get_asset`	Pulls all specific metadata and details for one particular Solana asset.
04	`get_assets_by_owner`	Lists every digital asset currently held by a specified wallet address.
05	`get_priority_fee_estimate`	Calculates the precise priority fee needed for a transaction, based on current network conditions.
06	`get_token_accounts`	Retrieves all token accounts associated with either a specific mint or owner address.
07	`get_transaction_history`	Pulls an enhanced and detailed log of activity for a given blockchain address.
08	`get_wallet_balances`	Gets the current token and NFT balances while also providing their equivalent USD value.
09	`get_wallet_funded_by`	Identifies the original source or funding origin of a specific wallet address.
10	`get_wallet_history`	Retrieves the complete and comprehensive transaction history log for an entire wallet.
11	`get_wallet_identity`	Resolves a Solana address to known entities, such as associated domain names.
12	`get_wallet_transfers`	Gathers all records of both incoming and outgoing token transfers for an account.
13	`parse_transactions`	Takes raw, technical transaction data and converts it into plain English, human-readable statements.

#	TOOL	DESCRIPTION
14	search_assets	Allows advanced filtering queries across the entire asset pool using criteria like collection or attributes.

See It in Action

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

U What is the current balance and USD value of the wallet 86W9...?

I've retrieved the balances for that wallet. It currently holds 15.5 SOL ($1,860), 500 USDC, and 3 NFTs. Total estimated value is $2,360.

U List all NFTs owned by 6p6... including their metadata.

I found 12 assets for this owner. Notable items include 'Mad Lad #402' and 'Famous Fox #123'. Would you like the full metadata for any of these?

U What is the recommended priority fee for a transaction right now?

Based on current network conditions, the 'Medium' priority fee is approximately 5,000 micro-lamports. For urgent transactions, 'High' is recommended at 25,000 micro-lamports.

Frequently Asked Questions

01 **Can Helius (Solana) MCP tell me the current dollar value of all NFTs?**

Yes, it can. Use get_wallet_balances to fetch both your token and NFT holdings, which are returned with their real-time USD valuations.

02 **What is the best way to check an address's full transaction history using Helius (Solana) MCP?**

You should use get_wallet_history for a complete log of every activity. If you only care about token movement, use get_wallet_transfers.

03 **How do I find out if an address was funded by a specific source?**

Run the get_wallet_funded_by tool. This specifically identifies and reports on the original funding source of the wallet, giving you crucial compliance data.

04 **Does Helius (Solana) MCP help me estimate gas fees?**

Absolutely. Use get_priority_fee_estimate to calculate the precise network fee required for your transaction, ensuring it goes through efficiently even during heavy congestion.

05 **Can I find assets based on their collection or creator using Helius (Solana) MCP?**

Yes. The search_assets tool allows you to filter the entire asset pool using advanced parameters like a specific collection name, creator, or trait.

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 `"helius-solana": { "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

Helius (Solana) 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 Helius (Solana). 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	Helius (Solana) MCP
Server ID	`019e38a7-c1ab-7131-8fd6-6ae8c2993165`
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/helius-solana.