MCP SERVER | NO CODE | CLOUD HOSTED

IPGeolocation MCP

Know the exact time and place of any IP address.

IPGeolocation provides deep global intelligence by resolving IP addresses and coordinates into precise location data, time zones, and local culture details. Your agent gets accurate GPS coordinates, ISP information, current local times with UTC offsets, financial currency symbols, and astronomical metrics like sunrise/sunset timings for any point on Earth.

A+ Quality Score 100/100

geolocation | ip-intelligence | timezone-data | astronomy-data | network-audit | localization

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/ipgeolocation-ip-intelligence-time

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.

IPGeolocation (IP Intelligence & Time) MCP

3 tools available
Cloud-hosted on Vinkius

You've got an application that needs to know where a user is—not just a city name, but the precise time they live in, what currency they use, and even if it's currently daylight saving time. This MCP connects your AI client directly to IPGeolocation's full intelligence suite. Instead of hitting separate APIs for location, network data, and timezones, you ask your agent one question: 'What is happening at this IP?' The result is a single, rich payload containing everything from GPS coordinates and ISP details to local currency symbols and planetary metrics like moon phases. Using Vinkius means you connect once and suddenly gain access to this entire global knowledge base through natural conversation. This lets developers build highly localized experiences and security teams conduct rapid network audits without leaving their chat interface.

Core Capabilities

01 — Determine Physical Location

Resolve any IP address or coordinate set into detailed geographic data, including city, country, GPS coordinates, and local currency.

02 — Calculate Local Timezone Details

Get the exact local time, UTC offset, and daylight saving status for a given global location.

03 — Audit Network Infrastructure

Extract deep details about an IP address's network origin, including the ISP name and organization type.

04 — Track Celestial Events

Retrieve astronomical data like sunrise, sunset, and moon phases for specific dates and locations.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/ipgeolocation-ip-intelligence-time** — connect your AI agent in three steps.

01 Subscribe to the MCP on Vinkius and enter your IPGeolocation API Key.

02 Invoke a tool in your agent using natural language prompts (e.g., 'What time is it in Tokyo?').

03 Your agent receives a structured data response containing geolocation, timezone, or astronomy details that you can use directly.

The bottom line is you get deep, real-world global data—location, network status, and calendar metrics—all through your chat interface.

Built For

This MCP is for the developer building a globally distributed product or the security analyst who needs instant context on suspicious traffic. You're tired of writing boilerplate code just to figure out if an IP address belongs in London, Tokyo, or somewhere else entirely.

Software Developer

Implementing user experiences that change based on location, like displaying local currency symbols or adjusting content for daylight saving time.

Security Analyst

Investigating suspicious IP addresses by getting immediate details on their geographic origin and the ISP responsible for them.

Data Engineer

Enriching large server logs or visitor datasets with accurate geolocation and astronomical metadata to improve analysis quality.

What Changes When You Connect

01 Pinpoint user location instantly. Use `get_location` to get GPS coordinates, city names, country codes, and even local currency symbols from an IP address—no extra lookups needed.

02 Handle global scheduling correctly. With the `get_timezone` tool, you immediately know the precise local time, UTC offset, and whether Daylight Saving Time is active in a user's region.

03 Improve data quality for logs. Data engineers can run network audits using this MCP to enrich server logs with accurate geolocation and ISP details, making analysis much smarter.

04 Build context-aware features. Use `get_astronomy` when your application needs to know if it's daytime or night at a given location, powering things like localized event scheduling.

05 Simplify complex logic. Instead of writing dozens of IF/THEN statements for time zones and countries, let your agent handle the complexity using structured data outputs.

Real-World Applications

Troubleshooting a Global Bug

A developer notices an app bug only happens during certain hours. They ask their agent to use `get_timezone` on the user's IP range, confirming that the issue correlates exactly with Daylight Saving Time transitions in a specific region.

Validating Server Traffic Origin

A security analyst receives an alert from a suspicious IP. They use this MCP to run `get_location` and immediately identify the country, GPS coordinates, and ISP name, narrowing down the investigation scope quickly.

Displaying Local Event Times

An e-commerce site needs to show a flash sale time. The developer asks their agent to use `get_astronomy` combined with `get_timezone` to calculate the precise local start/end times for all users, ensuring accuracy everywhere.

Localizing Pricing Audits

A data team is auditing pricing rules. They use this MCP's location features to check an IP address and pull not only the country but also the associated local currency symbol, ensuring localized pricing logic works.

Patterns to Avoid

Using Multiple APIs for One Goal

✗ AVOID

Calling a dedicated GeoIP tool, then another service for timezones, and finally calling a third service for currency data. This is slow, expensive, and hard to maintain.

✓ INSTEAD

Run the full intelligence sweep using this MCP. You can use `get_location` or `get_timezone` to pull all related details—like ISP names, coordinates, and timezones—in one clean operation.

Ignoring Timezone Nuances

✗ AVOID

Assuming a simple UTC offset is enough for scheduling. This fails during Daylight Saving Time transitions or when dealing with political timezone changes.

✓ INSTEAD

Always use `get_timezone`. It provides the full, accurate context including DST status and local time rules, which prevents your application from breaking at key dates.

Focusing Only on IP Address

✗ AVOID

Thinking you need an IP address to find location. Sometimes you only have coordinates (lat/long) or a specific city name.

✓ INSTEAD

Don't worry about the source. Use this MCP and specify your input as latitude/longitude when prompted, allowing you to run `get_astronomy` or other tools without an IP address.

The Right Fit

Use this MCP if your application logic hinges on knowing *where* a user is in the world—not just which continent. You need complex context: Does the data require local time zone rules? Are you tracking network origin? Do you need to display sunrise/sunset times or local currency symbols? If any of those are true, this MCP is essential. Don't use it if your only requirement is a simple country code lookup; another basic mapping tool might suffice. However, if you *do* need the full picture—ISP details combined with timezones and astronomy data—this singular source of truth saves massive development headaches.

Manual location checks are a huge time sink.

Right now, figuring out global context is a pain. You're dealing with separate tabs for GeoIP lookup, then switching to a calendar tool just to get the local date, and maybe hitting another API just to fetch the currency symbol. It's copy-pasting data between three or four different services just to answer: 'What time is this event happening, and how much will it cost?'

With this MCP, your agent handles all those hops for you. You ask a single question—for example, asking about an IP address in Brazil—and the system returns everything: GPS coordinates from `get_location`, local currency details, and the accurate time zone data needed to schedule the event correctly.

Get Location, Timezone, and Astronomy Data with One Call

The manual steps that disappear are: switching between geolocation services, writing complex date/time offset logic for every time zone, and managing separate calls just to get sun/moon phase data. You stop debugging cascading API failures.

Now, the complexity vanishes into a single, reliable data structure accessible through your agent's conversational flow. Your application gets global intelligence instantly.

IPGeolocation (IP Intelligence & Time) 3 Tools

Use these three tools to retrieve everything from a simple IP address lookup to complex astronomical calculations and precise timezone details.

#	TOOL	DESCRIPTION
01	`get_location`	Retrieves detailed geographic information, including GPS coordinates, for a given IP address.
02	`get_timezone`	Provides the precise local time, UTC offset, and daylight saving status for any specified region or coordinate set.
03	`get_astronomy`	Accesses astronomical data for a location, such as sunrise times, sunset times, and moon phases.

See It in Action

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

U Resolve the location for IP address '8.8.8.8'

Resolving IP '8.8.8.8'... This address is located in Mountain View, California, United States. It belongs to the 'Google LLC' ISP. GPS Coordinates: 37.4223, -122.0841. Local currency is USD ($).

U What is the current local time and UTC offset in Tokyo?

Current local time in Tokyo (Asia/Tokyo) is 10:45 PM (Tuesday). The UTC offset is +09:00. Daylight Saving Time is currently not active in this region.

U Show me the sunrise and sunset times for Lisbon today

Retrieving astronomy data for Lisbon... Today, the sun rises at 7:12 AM and sets at 6:45 PM. The solar noon occurs at 12:58 PM, and the total day length will be 11 hours and 33 minutes.

Frequently Asked Questions

01 **How does IPGeolocation get accurate timezone data using the get_timezone tool?**

The `get_timezone` tool returns precise time and zone information, including knowing if Daylight Saving Time is currently active. This prevents errors that plague systems relying on simple UTC offsets.

02 **Can I use IPGeolocation to find the ISP details of an IP?**

Yes. The location intelligence features allow you to extract detailed ISP information and organization names associated with any given IP request, which is useful for security audits.

03 **Do I need a physical address to use get_astronomy?**

No. You can run `get_astronomy` using direct latitude and longitude coordinates when an IP address isn't available, giving you sunrise/sunset data for that exact spot.

04 **What kind of data does the get_location tool provide?**

The `get_location` tool returns comprehensive geographic details. Expect city names, country codes, GPS coordinates, and even the local currency symbol associated with the IP's jurisdiction.

05 **Is this MCP better than using a simple database lookup for time zones?**

Yes. This MCP uses live global data sources. It accounts for historical rules, political changes, and DST shifts that static databases often fail to track accurately.

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 `"ipgeolocation-ip-intelligence-time": { "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

IPGeolocation (IP Intelligence & Time) 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 IPGeolocation (IP Intelligence & Time). 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	IPGeolocation (IP Intelligence & Time) MCP
Server ID	`019d75bb-95b0-7011-b827-27c876d42789`
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/ipgeolocation-ip-intelligence-time.

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