MCP SERVER | NO CODE | CLOUD HOSTED

NOAA Climate MCP

Analyze historical weather trends across decades.

NOAA Climate — Historical Weather Records provides access to the planet's largest archive of daily weather data, including temperature, precipitation, snow depth, and wind records for over 100,000 stations worldwide. You can retrieve detailed daily readings (GHCN-Daily), monthly averages (GSOM), or yearly summaries (GSOY) spanning decades. It also provides 30-year climate normals and station searches, making it the definitive source for historical climate science analysis.

A+ Quality Score 100/100

historical-data · climate-science · environmental-data · precipitation-records · temperature-analysis · data-archiving

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/noaa-climate-historical-weather-records

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.

NOAA Climate — Historical Weather Records MCP

5 tools available
Cloud-hosted on Vinkius

This MCP gives your agent direct access to NOAA's massive archive of global weather data. Forget sifting through dozens of academic databases or piecing together yearly reports. You can ask specific questions like, 'How did average rainfall change in Miami between 1980 and 2000?' The system pulls the raw historical records—daily temperature highs, precipitation totals, and snow accumulation—and formats them for immediate use. Whether you need a full year's worth of data or just the baseline thirty-year normal, this MCP handles it. You can pinpoint exact stations anywhere in the world and run analyses across daily, monthly, or annual scales. By connecting to Vinkius, you get all these climate tools under one roof, letting your AI client treat NOAA as a single, unified source for everything from local microclimates to continental trends.

Core Capabilities

01 — Fetch detailed daily weather readings

Retrieve specific measurements like maximum/minimum temperature and precipitation totals for any given date range and station.

02 — Calculate monthly climate averages

Generate aggregate summaries that provide average temperatures, total rainfall, or heating degree days for an entire month.

03 — Determine yearly climate trends

Get year-over-year data points, including annual temperature averages and extreme weather values.

04 — Establish baseline 'normal' conditions

Access the statistical 30-year baseline (1991–2020) to compare current readings against historical norms for a location.

05 — Locate weather stations globally

Search NOAA's network to find specific station IDs, names, and geographical coordinates needed for all other data calls.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/noaa-climate-historical-weather-records** — connect your AI agent in three steps.

01 First, use `search_stations` to get the precise ID and location name of the weather station you need.

02 Next, call one of the summary tools—like `get_monthly_summary` or `get_daily_data`—by passing that station ID along with your required date range and data type.

03 The MCP returns structured historical records (temperatures, precipitation, etc.) which your agent uses to build a final report or chart for you.

The bottom line is: instead of navigating complex government APIs, you just tell your AI client what time period and location you need, and it handles the data plumbing.

Built For

Forecasters, environmental consultants, agricultural planners, and academic researchers. This is for anyone who needs to prove a trend—be it drought severity, rising temperatures, or seasonal variability—using verifiable, decades-long datasets.

Climate Scientist

Compares current climate patterns against the 30-year baseline using `get_climate_normals` to identify anomalies.

Agricultural Planner

Analyzes historical rainfall and temperature cycles over multiple years using `get_yearly_summary` to determine optimal planting schedules for a region.

Insurance Risk Analyst

Retrieves detailed records of extreme weather events, such as maximum snow depth or total precipitation, spanning decades via `get_daily_data`.

What Changes When You Connect

01 You stop manually cross-referencing academic papers. By running `get_daily_data`, your agent pulls raw, verifiable daily records for temperature and precipitation directly from the source archive.

02 Comparing different time scales is simple. You can run a year's summary using `get_yearly_summary` and then immediately compare it to the established baseline using `get_climate_normals`. It’s all in one workflow.

03 Pinpointing location data used to be a headache. Now, just use `search_stations` first; you get the exact ID needed before calling any of the summary tools for accurate results.

04 You gain deep temporal insight. Instead of just knowing 'it rained last year,' your agent can quantify total precipitation and identify the wettest or driest months using `get_monthly_summary`.

05 The MCP handles complexity. You don't need to know how NOAA structures its data; you just ask for a trend, and the system figures out if it needs daily readings or yearly aggregates.

Real-World Applications

Investigating drought severity in agriculture

An agricultural planner needs to know how much total precipitation fell across the last five summers. They first use `search_stations` to get the county ID, then call `get_monthly_summary` for each of the past five years' months, giving them a clear trend line.

Assessing changes in coastal storm risks

A risk analyst wants to compare average high temperatures from 1950 versus today. They use `get_yearly_summary` for both periods and then run the data through their agent for immediate comparison against the 30-year climate normals.

Building a historical research paper on rainfall

A scientist needs raw daily rainfall totals. They use `search_stations` to locate all relevant points, then iteratively call `get_daily_data` across the entire time span and compile the massive dataset in one go.

Determining optimal building placement

An urban developer needs to check for extreme cold periods. They use `get_climate_normals` to establish a baseline low temperature, helping them determine if current local readings are dangerously outside the norm.

Patterns to Avoid

Assuming all data is available daily

✗ AVOID

Trying to get precipitation records for 100 years using only `get_daily_data` will fail because the dataset structure changes frequently and isn't designed that way.

✓ INSTEAD

For long-term trends, always check if a summary tool exists. If you need yearly comparison, use `get_yearly_summary`. For monthly averages over decades, rely on `get_monthly_summary`.

Skipping station identification

✗ AVOID

If the user just inputs 'Miami' into a data query without first running `search_stations`, the MCP will fail because it needs a specific NOAA ID (like USW00123456) to function.

✓ INSTEAD

Always start by using `search_stations` to confirm the exact station ID before calling any other tool. This ensures your data is correctly targeted.

Confusing 'normal' with 'average'

✗ AVOID

Thinking that the average of 1985-2024 will be accurate. The official baseline used for comparison is the standardized 30-year period.

✓ INSTEAD

When you want the accepted statistical standard, use `get_climate_normals`. This provides a clean, recognized baseline (1991–2020) that is distinct from simple arithmetic averages.

The Right Fit

Use this MCP if your primary goal is tracking environmental changes over time. You need to compare current conditions against historical performance—for instance, proving a trend in average temperature or total annual rainfall. If you only care about the weather for next Tuesday, using `get_daily_data` is overkill; checking simple local forecasts elsewhere will suffice. However, if your job requires quantifying long-term change (e.g., 'Did global warming increase drought severity?'), this MCP is essential because it allows you to pull data across different time granularities: use `search_stations` first, then choose between `get_daily_data` for micro-analysis, or `get_monthly_summary` and `get_yearly_summary` for macro-trends. Don't use this if your goal is just finding out what the weather will be next week; it’s a historical archive, not a forecast tool.

Climate data analysis used to involve massive spreadsheets and API juggling.

Before this MCP, analyzing climate trends meant downloading huge CSV files from NOAA's site. You'd spend hours cleaning the data—matching station IDs across different time periods, ensuring consistent date formats, and manually compiling summaries for comparison. It was a tedious cycle of copy-pasting, cross-referencing years, and arguing with spreadsheet formulas.

Now, you ask your agent to compare 1950 rainfall totals against today's readings. The MCP handles the data retrieval complexity: it uses `search_stations` to find the right ID, then calls `get_yearly_summary` for both dates. You don't see the API calls; you just get the clean comparison you need.

Get immediate climate baselines with NOAA Climate — Historical Weather Records MCP

The most time-consuming part was establishing a reliable 'normal.' You had to manually determine which 30-year window the data used. This process introduced human error and slowed down research considerably.

With this MCP, running `get_climate_normals` instantly gives you the standardized 1991–2020 baseline for any station. You get reliable scientific context in seconds, letting you focus on interpreting the results instead of cleaning the inputs.

NOAA Climate — Historical Weather Records (5 Tools)

These five tools allow your AI agent to interact with NOAA's full range of climate data, letting you analyze everything from daily temperature spikes to multi-decade averages.

#	TOOL	DESCRIPTION
01	get_daily_data	Pulls daily temperature, precipitation, snow depth, and wind records for specific dates at a given station.
02	get_monthly_summary	Generates monthly aggregates of average temperature, total rainfall, and heating degree days for trend analysis.
03	get_yearly_summary	Provides annual summaries detailing yearly averages and extreme values for long-term climate assessment.
04	get_climate_normals	Retrieves the standard 30-year statistical baseline (1991-2020) that defines 'normal' weather conditions for any location.
05	search_stations	Finds official NOAA station IDs and coordinates using a location name or bounding box, which are necessary inputs for all other tools.

See It in Action

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

U Get daily temperatures for Central Park, NYC in January 2024

📊 Central Park (USW00094728) — January 2024

DATE	MAX°C	MIN°C	PRECIP MM
Jan 1	4.4	-1.1	0.0
Jan 2	6.7	1.1	2.5
Jan 3	2.2	-3.3	0.0
...	...	...	...
Jan 31	7.8	0.0	8.1

Monthly avg high: 4.8°C. Monthly precip: 89mm.

U Show me the total monthly precipitation for Seattle in 2023.

Here is the GSOM data for Seattle (USW00024233) for 2023: Total annual precipitation was 31.5 inches. The wettest month was November with 6.2 inches, and the driest was August with 0.1 inches.

U What are the 30-year climate normals for Miami?

Based on the 1991-2020 normals for Miami International Airport: The average annual high is 84.4°F, the average low is 71.2°F, and the average annual precipitation is 61.9 inches.

Frequently Asked Questions

01 **How do I find a specific NOAA weather station ID using NOAA Climate — Historical Weather Records MCP?**

You must start by calling `search_stations`. This tool accepts location names or bounding boxes and returns the exact, necessary station IDs for every other data retrieval tool.

02 **What is the difference between using get_daily_data and get_monthly_summary with NOAA Climate — Historical Weather Records MCP?**

`get_daily_data` gives you granular records (max/min temp, precipitation) for every day. `get_monthly_summary` aggregates this data to give you averages and totals for the entire month, which is better for spotting general trends.

03 **Can I use NOAA Climate — Historical Weather Records MCP to compare temperatures across different years?**

Yes. You can use `get_yearly_summary` repeatedly across different decades (e.g., 1950 vs. 2020) to track yearly averages and extreme values.

04 **Does get_climate_normals cover all historical data?**

No, `get_climate_normals` provides the standardized statistical baseline (1991-2020). It is a reference point, not raw historical data.

05 **What if I need precipitation records for many different stations?**

First, you run `search_stations` to get the list of all required IDs. Then, your agent can iterate through that list, calling `get_daily_data` or `get_monthly_summary` for each ID and date range.

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 `"noaa-climate-historical-weather-records": { "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

NOAA Climate — Historical Weather Records 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

VINKIUS

INDEPENDENT PLATFORM DISCLAIMER

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	NOAA Climate — Historical Weather Records MCP
Server ID	`019d75de-768e-7362-83b4-57e2442dba59`
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/noaa-climate-historical-weather-records.