MCP SERVER | NO CODE | CLOUD HOSTED

WHOOP MCP

Translate Biometrics into Actionable Insights

WHOOP connects your wearable health data—sleep, recovery, strain, and workouts—directly into your agent. Get instant answers about body metrics, sleep stages, or training load without digging through apps or spreadsheets. This MCP lets you ask natural language questions about your full 24-hour cycle data, making advanced biometrics immediately accessible to any AI client.

A+ Quality Score 98.33/100

wearable-tech | sleep-tracking | recovery-metrics | heart-rate-monitoring | health-analytics | biometrics

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

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.

WHOOP MCP

11 tools available
Cloud-hosted on Vinkius

Managing physical performance is complex; it's not just about logging miles run or hours slept. It requires cross-referencing metrics like Heart Rate Variability (HRV), recovery scores, and deep sleep duration against your recent training load.

This MCP connects your WHOOP account to any AI client, letting you query this sophisticated health data through conversation. Your agent reads the raw numbers—whether it's tracking body measurements or analyzing a specific workout session—and synthesizes them into clear insights for you. You don't have to jump between apps to piece together what 'good recovery' really means.

It’s designed for people who need actionable data, not just graphs. When you connect this through Vinkius, your agent can access everything from general cycle summaries to specific sleep stages and even historical performance trends with simple prompts. You get a single source of truth about how your body is responding day-to-day.

Core Capabilities

01 — Analyzing Sleep Performance

Retrieve detailed records showing light, deep, REM sleep duration, and overall sleep performance for specific nights.

02 — Tracking Daily Recovery Status

Get a snapshot of recovery scores, resting heart rate, HRV, and sleep balance for any given 24-hour cycle period.

03 — Reviewing Training History

Pull specific workout records, detailing strain score, average heart rate zones, duration, and calories burned during a session.

04 — Monitoring Body Metrics

Access essential physical data like height, weight, and maximum heart rate alongside your performance metrics.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your required WHOOP Client ID, Secret, and Access Token.

02 Connect the credentialed MCP to your preferred AI client (Claude, Cursor, etc.).

03 Ask your agent a specific question, like 'What was my recovery score two days ago?' The MCP executes the request and returns the precise data.

The bottom line is that you just talk to your agent about your health metrics instead of navigating multiple dashboards or API endpoints.

Built For

This MCP serves anyone whose performance depends on understanding their physical limits and recovery. It's for the endurance athlete who needs to know if a heavy training day compromised sleep, the personal trainer monitoring client workload trends, or any bio-curious individual tracking deep biometrics over time.

Endurance Athlete

Using this MCP, you can correlate high workout strain with low recovery scores and poor sleep quality to adjust your training plan.

Physical Therapist / Coach

You monitor a client's trend data over weeks, looking for patterns where resting heart rate or HRV drop off following specific activity levels.

Bio-Curious Health Enthusiast

You track your sleep stages and overall body measurements to understand how lifestyle changes affect your deep restorative rest.

What Changes When You Connect

01 Instead of manually checking sleep logs, use `get_sleep` to pull all historical data for a date range and ask your agent: 'Show me my deep sleep percentage trend over the last month.'

02 Stop guessing about readiness. Use `get_cycle_recovery` to check immediate recovery scores (HRV, rest HR) right after an intense workout, letting you decide if you can push harder or need a rest day.

03 Cross-reference your physical health with activity using `get_workout`. You can ask: 'What was my average strain score during my run last Tuesday?'

04 When tracking long-term trends, use `get_recovery` and `get_sleep` together. Your agent compiles the narrative: 'My recovery scores drop every time I hit 30+ miles in a week.'

05 Need to check basic stats? The `get_body_measurement` tool ensures your agent has up-to-date metrics like weight or max heart rate for accurate comparisons.

Real-World Applications

Diagnosing Overtraining Syndrome

A coach needs to know if a client's performance is declining due to poor rest. The agent uses `get_cycles` and compares the resulting strain scores against the historical low recovery metrics pulled via `get_cycle_recovery`, immediately flagging potential overtraining patterns.

Planning Travel-Related Workouts

You're packing for a trip and need to know your baseline fitness. You ask the agent, 'Based on my last 10 workouts via `get_workouts`, what is my average sustained heart rate zone performance?' This helps you plan appropriate activity levels.

Reviewing Sleep Debt

You suspect your sleep quality has worsened. You use `get_sleep` for the past two weeks and ask: 'Identify any nights where my REM sleep was less than 1 hour.' The agent analyzes the data points from multiple records.

Post-Injury Workload Management

A physical therapist needs to ensure a patient isn't overdoing it. They use `get_cycle` and ask: 'How was my recovery on days I used the rehab machine?' The agent compares daily activity logs against core biometrics.

Patterns to Avoid

Treating data points as standalone facts

✗ AVOID

Asking your agent, 'What was my sleep score on July 1st?' and getting only a single number without context.

✓ INSTEAD

Instead, ask the agent to compare two metrics: 'How did my deep sleep percentage (using `get_sleep_by_id`) change when I hit a high strain workout (checked with `get_workout`)?' Context is everything.

Trying to predict future performance

✗ AVOID

Asking, 'Will I run a marathon next year?' The agent will give vague advice based on limited data.

✓ INSTEAD

Focus the query on historical trends. Ask: 'What is my average weekly total strain over the last quarter using `get_workouts` and `get_cycles`?' Stick to what the data shows.

Ignoring time constraints

✗ AVOID

Asking, 'Give me all my stats.' This results in a massive wall of text with no actionable takeaways.

✓ INSTEAD

Be specific. Limit the scope: 'Show me my recovery data for the last 14 days using `get_recovery` and summarize any dips below 60%.'

The Right Fit

Use this MCP if your primary need is to synthesize complex, disparate biometrics (sleep stages, strain scores, HRV) into natural language answers. You should use it when you need your agent to *compare* metrics across time or correlate two different data types—for example, linking a high workout strain score to the following night's low recovery score.

Don’t use this MCP if: 1) You need raw, unaggregated sensor streams (e.g., every single heart rate reading per second). This tool provides summarized, structured API endpoints. 2) Your goal is simply data storage or visualization without asking a question. If you just want to see a chart of your weight over time, a dedicated charting service might be better.

This MCP excels at the 'Why'—it answers why your recovery was

low, or why your sleep wasn't restorative. It requires specific questions that force the agent to cross-reference tools like `get_cycle_sleep` and `get_cycle_recovery` against each other.

The struggle of assembling a complete performance picture

Right now, figuring out your body's true state is manual work. You might log a workout in one app, check sleep data on another dashboard, and look up recovery scores via a third source. Then you open a spreadsheet to try and manually correlate if the effort justified the rest.

With this MCP, you skip the dashboards entirely. You just tell your agent what you want to know —'Was my body ready for that run?' The system handles all the cross-referencing using tools like `get_workout` and `get_cycle_recovery`, giving you one immediate answer instead of five tabs to check.

WHOOP MCP: Performance Data in Conversation

The biggest time sink is the constant context-switching. You spend minutes clicking through date ranges, exporting CSVs, and pasting everything into a note just to write down three key takeaways for your coach.

Now, you talk to your agent. It accesses all the relevant data—whether it's pulling specific sleep stages via `get_sleep_by_id` or reviewing multi-day trends with `get_cycles`. The insight comes directly back in plain English.

WHOOP MCP: 11 Health Metric Tools

Use these eleven specific tools to analyze every aspect of your WHOOP data, from daily recovery scores to historical workout performance.

#	TOOL	DESCRIPTION
01	`get_body_measurement`	Retrieves current physical metrics like height, weight, and maximum heart rate associated with your profile.
02	`get_cycle`	Gets a complete 24-hour data summary, including sleep, recovery, strain, and general heart rate information for one specific cycle ID.
03	`get_cycle_recovery`	Fetches detailed recovery metrics, including HRV and the calculated score, for a single WHOOP cycle period.
04	`get_cycle_sleep`	Provides deep analysis of sleep data for a specific cycle, detailing stages like light, deep, REM, and overall performance percentage.
05	`get_cycles`	Retrieves WHOOP cycle data over a date range, allowing you to paginate and review many days of combined recovery, strain, and sleep metrics.
06	`get_profile`	Verifies your connection and returns basic user profile information needed for other endpoints.
07	`get_recovery`	Gets recovery data across a specified date range, useful for tracking trends over weeks or months.
08	`get_sleep`	Retrieves sleep data over a date range, allowing you to view historical performance and stage metrics.
09	`get_sleep_by_id`	Focuses on one specific night's sleep record, giving details on stages, disturbances, and respiratory rate.
10	`get_workout`	Pulls a single workout session by ID, providing strain score, heart rate zones, duration, and calories burned.
11	`get_workouts`	Gets all workout data within a specified date range, supporting pagination for comprehensive history review.

See It in Action

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

U Show me my recovery score from today.

Today's recovery: 78% (Good). Resting HR: 52 bpm, HRV: 68ms, sleep balance: 8h 12m. You're well recovered and ready for high strain today.

U How did I sleep last night?

Last night's sleep: 7h 45m total. Sleep performance: 85%. Light sleep: 3h 20m, Deep sleep: 2h 10m, REM: 1h 45m, Awake: 30m. Respiratory rate: 14 breaths/min. 2 disturbances.

U Show me my workouts from this week.

This week's workouts: 1. Running (45min, strain 14.2, avg HR 152, 520 cal) — 2. Weight Training (60min, strain 12.8, avg HR 138, 410 cal) — 3. Cycling (30min, strain 10.5, avg HR 145, 350 cal). Total weekly strain: 37.5.

Frequently Asked Questions

01 **How do I track long-term recovery data using the WHOOP MCP?**

You use the `get_recovery` tool, which lets you define a date range. This allows your agent to pull all your recovery scores over weeks or months so you can spot trends.

02 **Can I check my sleep stages for last night with WHOOP MCP?**

Yes. The `get_sleep` tool provides historical data, and the specialized `get_cycle_sleep` tool gives a deep dive into light, deep, REM, and awake time for any specific cycle.

03 **What kind of workouts can I analyze with WHOOP MCP?**

The `get_workouts` tool pulls comprehensive data across a date range. You get details like strain score, average heart rate zones, duration, and calories burned for each session.

04 **Do I need to use `get_profile` every time I connect WHOOP MCP?**

No, you only run the `get_profile` tool if your agent needs to verify basic user details or authentication status before calling other endpoints.

05 **Can I see my body measurements and recovery data together? (WHOOP MCP)**

Yes. You can combine information by using `get_body_measurement` for static stats, then correlating those with dynamic readings from `get_cycle_recovery` to get a full picture.

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 `"whoop": { "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

WHOOP 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 WHOOP. 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	WHOOP MCP
Server ID	`019d849d-ccc6-7358-844d-2ba0fe0302a7`
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/whoop.