MCP SERVER | NO CODE | CLOUD HOSTED

Alexa Smart Home MCP for AI Agents

Manage lights and climate control across smart properties

Alexa Smart Home MCP allows your AI agent to manage nearly every smart device connected to Amazon Alexa, from lights and speakers to thermostats and sensors. You can list all devices, check current status (like temperature or volume), and issue commands like turning things on/off, adjusting brightness, or changing room assignments.

A+ Quality Score 98.33/100

smart-home | automation | device-management | home-assistant | api-integration | sensor-data

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/alexa-smart-home

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.

Alexa Smart Home MCP

16 tools available
Cloud-hosted on Vinkius

Managing a modern home means juggling multiple apps and manual controls. This MCP lets your AI agent do the heavy lifting. Instead of opening five different manufacturer apps to dim lights, check the living room temperature, and adjust the speaker volume, you just talk to your assistant. Your agent talks directly to Alexa's infrastructure through this connection. You can get a full list of every device in the house, pull detailed information on specific units, and execute precise actions like setting brightness levels or updating friendly names across multiple locations.

It’s not just about turning things on and off; you can automate complex routines. Need to know if the HVAC system is heating enough? Your agent checks the thermostat state and reads live temperature sensor data. The whole process, from reading current settings to executing a change, happens instantly. By connecting this MCP through Vinkius's catalog, your AI client gains expert-level control over your entire smart property, making manual oversight obsolete.

Core Capabilities

01 — Surveying Connected Devices

List all connected Alexa devices and retrieve detailed information for any specific unit in the house.

02 — Controlling Power and Light

Turn individual devices on or off, set them to a specific brightness level, or adjust their current power state.

03 — Managing Audio and Climate

Check the speaker volume status and get the current operating mode and target temperature from connected thermostats.

04 — Locating and Organizing Devices

Update a device's friendly name or assign it to a different physical room in your home setup.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/alexa-smart-home** — connect your AI agent in three steps.

01 First, subscribe to this MCP on Vinkius and obtain the necessary OAuth 2.0 access token via Amazon's Login with Amazon.

02 Next, connect that token to your preferred AI client (like Cursor or Claude). This gives your agent the authority to issue commands against your Alexa account.

03 Finally, you ask your agent a request—for example, 'It's too dark in the office.' Your agent translates this into specific calls, adjusting brightness via tools like adjust_brightness and set_brightness.

The bottom line is: once authenticated, your AI client acts as a unified control layer for all your Alexa-connected smart devices.

Built For

This MCP is essential for anyone who manages physical space and needs granular, real-time control over infrastructure. Property managers dealing with multiple units, hospitality operators running hotels, or even tech-savvy homeowners automating complex routines benefit most.

Property Manager

Checks device power states across several properties to ensure lights and climate controls are active for incoming guests.

Hospitality Operator

Reads thermostat state or adjusts brightness levels in hotel rooms remotely, optimizing guest comfort and energy use simultaneously.

Home Automation Enthusiast

Integrates Alexa devices into larger scripts, using tools like update_device_room to create sophisticated, multi-step routines.

What Changes When You Connect

01 Real-time status checks: Instead of guessing, your agent calls get_temperature_sensor or get_thermostat_state to provide exact climate data for decision making.

02 Centralized device management: You can list all devices using list_alexa_devices and then use update_device_room to assign them accurately without opening multiple apps.

03 Fine-tuned lighting control: The MCP supports both setting specific brightness levels (set_brightness) and making relative adjustments (adjust_brightness), giving granular light management.

04 Unified power control: You can reliably turn devices on or off using turn_on_alexa_device or turn_off_alexa_device, regardless of whether it's a plug, switch, or lamp.

05 Improved user experience: Easily rename any device with update_alexa_device_name so your agent uses clear language like 'kitchen speaker' instead of a complex model number.

Real-World Applications

Pre-Departure Security Check

A property manager asks their AI agent to prepare for vacancy. The agent first calls list_alexa_devices, then sends commands using turn_off_alexa_device and set_volume on all listed units, ensuring nothing is left running or playing.

Adjusting a Meeting Room's Ambiance

During a conference call, a user asks their agent to make the room darker. The agent uses get_brightness_state, then calls adjust_brightness and set_volume on the lights and speakers, optimizing the mood for the meeting.

Troubleshooting HVAC Issues

A guest reports the room is too cold. The agent checks the current state using get_thermostat_state and then reads ambient data with get_temperature_sensor to report back if the system needs adjusting.

Onboarding a New Tenant

The agent helps set up new hardware by first calling list_alexa_devices, identifying unused endpoints, and then guiding the user through setting the correct friendly name using update_alexa_device_name.

Patterns to Avoid

Assuming a device is connected

✗ AVOID

Telling your agent to check the temperature in the guest wing without first confirming that specific sensor ID exists, leading to an error.

✓ INSTEAD

Always start by listing devices using list_alexa_devices. Use this initial inventory to confirm the exact endpoint IDs before attempting any read or write operation like get_temperature_sensor.

Confusing 'forget' with 'deregister'

✗ AVOID

Calling forget_alexa_device when you just need to temporarily disconnect a device for maintenance, which would require re-pairing it entirely.

✓ INSTEAD

If the physical device is still fine but needs temporary removal from your active account view, use deregister_alexa_device. Only use forget_alexa_device if you are sure it needs total erasure.

Ignoring room assignments

✗ AVOID

Asking the agent to turn off a device by name when that device might be used in multiple rooms, leading to unintended power loss.

✓ INSTEAD

If context matters, always use update_device_room first. This ensures your AI client knows exactly which physical unit you are referencing before issuing any command like turn_off_alexa_device.

The Right Fit

Use this MCP if your workflow requires granular control over multiple distinct smart home subsystems, such as lighting, climate, and audio. Specifically, use it when you need the AI agent to read current statuses (like get_speaker_state or get_thermostat_state) before taking action.

Don't use this if you are only working with a single brand of device that has its own dedicated API, as that might be cleaner. Also, don't rely on it for non-connected services; this MCP requires the physical devices to already be registered within your Alexa account. If all you need is simple scheduling (e.g., 'lights off at 10 PM'), a basic automation routine tool may suffice—you won't need the full power of multiple reading and writing capabilities.

Alexa Smart Home MCP: Managing Device Status in Property Operations

Right now, checking property status is a manual nightmare. You log into one dashboard for climate control, another for lighting, and maybe a third just to check speaker volume. You spend time clicking through menus just to answer simple questions like 'Is the AC running in Unit 3?' or 'Are all lights off?'

With this MCP, your agent aggregates that data instantly. Instead of jumping between tabs, you ask one question ('What's the status of unit three?'), and the AI client pulls together reports on everything—from get_thermostat_state readings to power states for every single light fixture.

Alexa Smart Home MCP: Controlling Device Settings in Property Operations

The biggest pain point is executing complex changes. If you need to dim the lights and adjust the temperature, you'd issue two separate commands across different interfaces. You risk

Now, your agent coordinates everything. It uses multiple tools—like set_brightness followed by turn_on_alexa_device—in sequence to execute an entire scenario flawlessly. Your system moves

missing a step or getting mixed up on which device ID belongs where.

from fragmented manual clicks to unified, natural language commands.

Alexa Smart Home MCP: 16 Tools for Device Status & Control

Use these tools to read device status, update settings, or control the power flow of any Alexa-connected smart home appliance.

#	TOOL	DESCRIPTION
01	adjust_brightness	Increases or decreases the light brightness relative to its current setting.
02	deregister_alexa_device	Removes the tracking of a device from your account without resetting the physical hardware.
03	forget_alexa_device	Completely removes (forgets) an Alexa-connected device, requiring re-setup if you want to use it again.
04	get_brightness_state	Reads the current light brightness level of a connected light fixture.
05	get_alexa_device	Retrieves detailed information about a specific Alexa-connected device using its unique ID.
06	get_power_state	Checks if an Alexa-connected device is currently on or off.
07	get_speaker_state	Reads the current volume level of a connected Amazon speaker device.
08	get_temperature_sensor	Reads the current ambient temperature from a connected sensor unit.
09	get_thermostat_state	Retrieves the operating mode and target temperature of an Alexa-connected thermostat.
10	list_alexa_devices	Shows all connected smart home devices, providing names and necessary IDs for other commands.
11	set_brightness	Sets the desired brightness percentage for an Alexa-connected light.
12	set_volume	Sets the desired playback volume on an Alexa-connected speaker.
13	turn_off_alexa_device	Shuts down an Alexa-connected device, useful for turning off lights or plugs.
14	turn_on_alexa_device	Turns a specific, power-controllable Alexa device on.
15	update_alexa_device_name	Changes the visible, friendly name for a specific device in your account.

#	TOOL	DESCRIPTION
16	update_device_room	Moves an Alexa-connected device record to a different room or unit assignment.

See It in Action

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

U I'm leaving the house for the night; can you secure the unit?

Unit Security Checklist:

- **Lights:** All 12 devices are confirmed OFF. (Total power savings: High)
- **Climate:** Thermostat set to ECO mode with a target of 16°C.
- **Speakers:** Volume confirmed at 0. No active audio playing.

Please confirm if you need any specific device renamed or moved before I lock the system.

U The conference room is too dim and the speaker isn't loud enough.

Room Adjustment Complete:

- **Lighting:** Brightness increased to 85%. (Previously at 40%).
- **Audio:** Speaker volume set to 6/10.
- **Status:** All lights are now fully operational. The system is ready for the meeting.

U What's going on with the smart plug in the garage?

Garage Device Status:

METRIC	VALUE
Current Power State	OFF
Last Activity Time	2 hours ago
Assigned Room	Garage Unit A

It looks like the smart plug is currently off, and its last recorded activity was two hours ago. You can easily turn it on using a simple command.

Frequently Asked Questions

01 **How does the Alexa Smart Home MCP help manage lights in my building?**

It gives your AI agent granular, real-time control over every connected light. You can not only turn them on or off but also adjust brightness precisely using tools like set_brightness and adjust_brightness for perfect ambiance.

02 **Do I need to know complex API IDs when using the Alexa Smart Home MCP?**

No. You don't. The MCP handles the technical details behind the scenes. You just tell your agent what you want done—like 'dim the bedroom lights'—and it manages the necessary calls.

03 **Can the Alexa Smart Home MCP help me with temperature and HVAC issues?**

Yes, absolutely. Your agent can check the current state using get_thermostat_state and read ambient data via get_temperature_sensor, letting you know if your heating or cooling system is working correctly.

04 **Is this MCP better than just controlling devices manually through Alexa?**

The biggest difference is context. Instead of executing a single command, the AI client can run a sequence—for example, turning off lights AND setting the thermostat to eco mode all in one conversation.

05 **What if I rename a device using this MCP, will other systems notice?**

Yes. Using update_alexa_device_name makes sure that every integrated system your agent talks to uses the new, correct name instantly. It keeps your entire automation setup consistent.

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 `"alexa-smart-home": { "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

Alexa Smart Home 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 Alexa Smart Home. 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	Alexa Smart Home MCP
Server ID	`019d754b-f437-7092-851f-29c8a887b9fb`
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/alexa-smart-home.