MCP SERVER | NO CODE | CLOUD HOSTED

Linode (Akamai) MCP

Manage infrastructure from conversation.

Linode (Akamai) MCP lets your AI agent handle all cloud infrastructure tasks—from spinning up new virtual private servers to managing complex Kubernetes clusters. You can list, create, delete, and update compute instances, plus audit account details or user access using simple conversation. It gives you full control over your Linode environment right from your chat interface.

A+ Quality Score 100/100

vps | kubernetes | cloud-computing | server-management | infrastructure-provisioning | container-orchestration

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

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.

Linode (Akamai) MCP

10 tools available
Cloud-hosted on Vinkius

Managing a cloud setup used to mean jumping between dozens of dashboards: the billing page here, the networking console there, the cluster management portal somewhere else. Now, you talk to your AI client and it handles the complexity. This MCP lets you manage every aspect of your Linode infrastructure using natural language commands. Need a new staging environment? Just ask. Want to check who has access or if an old database instance is running hot? Ask that too. It's all connected. By connecting this resource via Vinkius, you get instant visibility into account details and the power to provision compute resources, deploy Kubernetes engines, and manage user accounts without ever logging into a web console. You treat your entire cloud environment like another function call in your code.

Core Capabilities

01 — Manage virtual private servers

You can list all current computing instances across different regions or quickly create and update a server with specific images.

02 — Provision Kubernetes clusters

The MCP handles deploying, listing, and retrieving details for your Linode Kubernetes Engine (LKE) clusters.

03 — Audit account security and users

Retrieve overall account billing information or list all authorized user accounts to verify permissions.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe this MCP and provide your Linode Personal Access Token.

02 Next, tell your AI client exactly what you want—for example, 'List all my compute instances in the US-East region' or 'Create a new LKE cluster'.

03 The agent executes the request against Linode's APIs and returns structured data about the resources to your chat window.

The bottom line is that you interact with your cloud provider using conversational prompts instead of complex web forms and CLI commands.

Built For

This MCP is built for the infrastructure owner, the developer who needs a quick staging environment, or the architect auditing resource access. It's for anyone whose day involves moving data between different console tabs.

DevOps Engineer

They use this to quickly check cluster statuses across multiple environments or spin up temporary testing resources without leaving their main terminal.

Full-stack Developer

They manage backend infrastructure—like creating a new database instance or updating existing ones—directly from the code editor while building an application.

Cloud Architect

They audit account resources and user access, verifying who has what permissions and ensuring resource allocation matches business needs.

What Changes When You Connect

01 You maintain security and visibility by using the `list_users` tool to see who has access, without digging through complex user management dashboards. It's all a single query away.

02 Instead of manually logging into multiple consoles, you use the MCP to list all compute instances or create new ones with specific hardware types. The whole process is conversational.

03 Need a scalable application environment? You can deploy and manage Kubernetes clusters using `create_lke_cluster` simply by describing what you need in plain English.

04 You get instant oversight on account status by running `get_account`. This provides quick access to billing data and resource summaries, saving you from navigating the main dashboard.

05 If a server needs tweaking, you don't have to delete it. You can use the `update_linode` tool to resize or change an instance's image without downtime.

Real-World Applications

Spinning up a temporary testing environment

A developer needs a test server immediately. They prompt their agent: 'Create a g6-standard-1 Linode in the eu-central region with Ubuntu 22.04.' The agent uses `create_linode`, and within minutes, the resource is provisioned and ready for testing.

Auditing user permissions after an employee leaves

The security team suspects a former contractor left backdoor access. They use the MCP to run `list_users` and immediately check every authorized account, ensuring no rogue keys or users remain active.

Debugging cluster connectivity issues

A site is reporting high latency on its main application. The architect asks the agent to use `get_lke_cluster` and `list_linodes`. This returns real-time data, pinpointing whether the issue lies with the K8s configuration or a specific compute node.

Scaling up production capacity quickly

Traffic suddenly spikes. Instead of manually checking current resources, the DevOps engineer asks to `list_linodes` and then prompts for an update on the largest instance using `update_linode`, scaling it instantly without leaving their workflow.

Patterns to Avoid

Treating it like a basic file listing

✗ AVOID

Thinking you just need to list names and IPs. This wastes time because the AI client can't give status or history.

✓ INSTEAD

Don't just ask for 'list all servers.' Instead, prompt: 'List all compute instances, showing their status and associated region.' Use `list_linodes` for concrete details.

Trying to fix things without context

✗ AVOID

You only know an instance is down. You can't figure out *why* it's down by just asking 'What's wrong?'

✓ INSTEAD

To troubleshoot, you must first run `get_linode` on the specific ID and then check account details using `get_account`. This gives the full context.

Over-provisioning resources by accident

✗ AVOID

You accidentally spin up five test clusters because you forgot to tell the agent which ones were temporary.

✓ INSTEAD

Before creating anything, always run `list_lke_clusters` and verify that existing clusters are listed. Then use `create_lke_cluster` with strict naming conventions.

The Right Fit

Use this MCP if your core workflow involves managing the lifecycle of cloud resources: provisioning, scaling, or auditing access rights. If you need to interact with compute instances (VPS) or container orchestration (Kubernetes), this is the tool. Don't use it if you are only managing billing records—use a dedicated billing connector instead. Similarly, don't use it just for file storage; that requires an object storage MCP. This MCP excels at infrastructure state management: read current status (`list_linodes` , `get_account`), and write new state (`create_linode` , `update_linode`). If your need is purely reporting (e.g., 'show me my last month's usage graph'), you'll need an analytics tool, not this infrastructure control MCP.

The Pain of Dashboard Hopping

Today, checking your cloud environment requires a painful ritual. You open the main web dashboard to list all nodes. Then you click into the Kubernetes section just to see cluster versions. If you need to create a new test server, you have to navigate away entirely, fill out three forms, and wait for confirmation emails. It’s slow, it's error-prone, and it takes too many context switches.

With this MCP, all that happens in one chat thread. You tell your agent: 'I need a new staging cluster with five nodes.' The system handles the multi-step deployment—the listing, the configuration checks, and the final creation command—and delivers confirmation instantly. It turns complex infrastructure management into simple conversation.

Deploying Infrastructure via Linode (Akamai)

Manual deployment requires logging in, selecting the correct region from a dropdown, choosing an image, and then hitting 'Confirm.' If you miss one step or select the wrong size, your entire process stalls. Furthermore, tracking which resources were created by whom is often hidden deep within logs.

Now, you just talk to your agent. You say: 'Please provision me a new compute instance in ap-south using Ubuntu 22.04.' The tool executes `create_linode`, confirms the ID and region, and provides the status update—all before you finish reading the sentence.

Linode (Akamai) MCP with 10 Tools

These tools give your agent the power to list, create, update, and delete everything related to your Linode cloud infrastructure.

#	TOOL	DESCRIPTION
01	`create_linode`	Creates a brand new virtual computing instance on Linode.
02	`create_lke_cluster`	Deploys an entire Linode Kubernetes Engine (LKE) cluster in your account.
03	`delete_linode`	Permanently removes a specified virtual computing instance from your cloud environment.
04	`get_account`	Retrieves key details about your overall Linode account status and billing information.
05	`get_linode`	Displays all specific details for a single, existing computing instance you own.
06	`get_lke_cluster`	Fetches the detailed status and configuration of one specific Kubernetes cluster.
07	`list_linodes`	Lists every compute instance currently running across all your Linode regions.
08	`list_lke_clusters`	Retrieves a comprehensive list of all Kubernetes clusters deployed in your account.
09	`list_users`	Lists all authorized users and their access permissions within the Linode account.
10	`update_linode`	Modifies an existing computing instance, such as changing its size or region.

See It in Action

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

U List all my current Linode instances.

I've retrieved your Linodes. You have 3 active instances: 'web-server-prod' (ID: 12345) in us-east, 'db-master' (ID: 12346) in eu-central, and 'staging-app' (ID: 12347) in ap-south.

U Create a new g6-standard-1 Linode in the us-east region with Ubuntu 22.04.

Provisioning your new Linode... The instance has been created with ID 54321. It is currently booting in the us-east region with Ubuntu 22.04 LTS.

U Show me the details of my Kubernetes clusters.

Fetching LKE clusters... I found one cluster: 'production-cluster' (ID: 9876). It's running K8s version 1.25 in us-east with 3 nodes in the default pool.

Frequently Asked Questions

01 **How do I list all my Linode compute instances using the Linode (Akamai) MCP?**

You use the `list_linodes` tool. This function returns a complete list of every virtual private server you own, including their IDs and current regions.

02 **Can I create a Kubernetes cluster with Linode (Akamai) MCP?**

Yes, use the `create_lke_cluster` tool. This deploys an entire LKE cluster, handling node pool configuration so you can start running containers right away.

03 **What is the best way to check my account billing details with Linode (Akamai) MCP?**

Use `get_account`. This tool retrieves your overall account status and key financial metrics, giving you a clean overview of resource consumption.

04 **How do I delete an old server using the Linode (Akamai) MCP?**

You run the `delete_linode` function. Make sure you have the correct instance ID first, then confirm deletion to prevent accidental data loss.

05 **What tools are available in the Linode (Akamai) MCP for user management?**

You can manage users and permissions using `list_users`. This tool lets you see who has access rights across your entire cloud account.

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 `"linode-akamai": { "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

Linode (Akamai) 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 Linode (Akamai). 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	Linode (Akamai) MCP
Server ID	`019e38b8-4b09-7160-84e4-95f141cf9159`
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/linode-akamai.