MCP SERVER | NO CODE | CLOUD HOSTED

Keycloak MCP

Manage identity and access control without the console.

Keycloak MCP manages identity and access control directly through your AI agent. You'll use this to audit security realms, create or delete users, manage groups, and configure OIDC/SAML clients without clicking a single button in the console.

F Quality Score 3.6/100

iam | authentication | authorization | sso | keycloak-admin

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

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.

Keycloak MCP

34 tools available
Cloud-hosted on Vinkius

This connector gives you full command over complex Identity and Access Management (IAM) processes. Instead of navigating through multiple Keycloak consoles or writing repetitive scripts, you talk to your agent about what needs fixing—whether it's deleting an orphaned client record or auditing who changed a realm setting last week. You can manage the core security infrastructure by simply asking for it. The system handles the complex API calls needed to update user credentials, assign roles, and force global logouts across entire realms. When you connect this MCP through Vinkius, your agent gets access to thousands of other specialized tools, so you stay in one place to handle everything from user lifecycle management to advanced security auditing.

Core Capabilities

01 — Manage User Accounts

Create new users, update existing details, reset passwords, or delete accounts across different realms.

02 — Administer Security Realms

List and import entire security environments (realms), or audit changes using the list_admin_events tool.

03 — Handle Client Applications

Create, read, update, or delete client applications, and instantly regenerate forgotten secrets like `get_client_secret`.

04 — Define Roles and Groups

Organize your security structure by creating top-level groups, assigning roles at the realm level, or managing user group memberships.

05 — Control Sessions and Access

Force a global logout across an entire realm (`logout_all_users`) to mitigate immediate security threats.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP on Vinkius and provide your Keycloak Base URL along with a valid Admin Access Token.

02 Second, point your AI client (Claude, Cursor, or any compatible agent) to the newly connected Keycloak data stream.

03 Third, prompt your agent using natural language—for example, 'List all users in the staging realm and reset John Doe's password.' — and watch it execute the necessary commands.

The bottom line is that you get to run complex identity management tasks through conversation instead of console clicks or code deployments.

Built For

This MCP is built for engineers and security staff who are tired of context switching between a ticketing system, the main application dashboard, and multiple API consoles. It's perfect for the DevOps engineer who needs to quickly audit permissions during a deployment or the Security Admin who has to perform emergency password resets at 2 AM.

DevOps Engineer

Using this MCP, they run `list_admin_events` and check client configurations right from their terminal flow to ensure deployment readiness.

Security Administrator

When a breach is suspected, they immediately use the agent to call tools like `logout_all_users` or perform targeted password resets via `reset_user_password`.

Backend Developer

They set up test environments by calling `create_client` and generating necessary credentials using `get_client_secret` without leaving their IDE.

What Changes When You Connect

01 You eliminate context switching. Instead of jumping between Keycloak's user list, group manager, and client config pages, you ask your agent to handle it all in one chat window.

02 Instant security response. If you suspect a breach, calling `logout_all_users` through the MCP instantly terminates every active session across the entire realm—no manual work required.

03 Never lose credentials again. With tools like `get_client_secret` and `regenerate_client_secret`, your agent retrieves or updates sensitive keys immediately upon request.

04 Full audit trail visibility. Use `list_admin_events` to get a clean, natural language summary of who changed what and when across the entire security infrastructure.

05 Efficient user lifecycle management. You can quickly call `create_user` or `delete_user`, ensuring accounts are provisioned or decommissioned exactly when needed.

Real-World Applications

The developer needs to onboard a new service.

A backend developer runs into an issue because the staging microservice is missing necessary permissions. They prompt their agent: 'I need to create a client for the payments service and assign it read-only access.' The agent calls `create_client` and then uses tools like `list_roles` and `update_client` to secure the connection, all in one go.

Security audit detects stale accounts.

A security admin finds a list of users who haven't logged in for months. They ask their agent to 'List all inactive user accounts older than 90 days and delete them.' The agent calls `list_users` and then executes multiple `delete_user` commands, completing the cleanup cycle.

A key application is compromised.

The ops team realizes an entire segment of users' access might be at risk. They immediately instruct their agent to 'Force a global logout across all production realms.' The agent calls `logout_all_users`, mitigating the threat instantly without any manual intervention.

Restructuring user teams.

The HR team requires that all members of the Marketing department be moved into a new group. An admin prompts: 'Create a group called 'Mktg-V3' and add every user currently in the old Mktg group.' The agent handles `create_group` and updates membership via tools like `update_user`.

Patterns to Avoid

Manual credential retrieval

✗ AVOID

A developer manually navigates the Keycloak UI, finds the client ID, clicks 'Secrets', copies the secret key, and pastes it into a config file. This is slow and prone to copy/paste errors.

✓ INSTEAD

Instead, use the agent to call `get_client_secret` directly with the client name. The agent handles the secure retrieval of the secret in one step.

Ad-hoc user updates

✗ AVOID

An administrator logs into the UI, finds a specific user by ID, clicks 'Edit,' and changes their status manually. If there are fifty users to update, this is agonizing.

✓ INSTEAD

Use `list_users` first to get all target accounts, then instruct your agent to run bulk updates using multiple calls to `update_user` for the entire list.

Complex role assignment

✗ AVOID

You need to assign three different roles (Admin, Viewer, Editor) to a user while also ensuring they are part of two specific groups. This requires navigating separate tabs and menus.

✓ INSTEAD

Prompt the agent: 'Assign Admin and Viewer roles to John Doe and add him to both the Sales and Engineering groups.' The agent coordinates `list_roles`, `get_role`, and `update_user` in sequence.

The Right Fit

Use this MCP if your primary pain points involve Identity, Authentication, or Authorization (IAM). If you need to audit who can do what, manage user lifecycles (create/delete), or modify client settings, this is the right tool. However, don't use it if you only need simple logging—if you just want a feed of system events without needing to change anything, general logging tools are better suited.

Also, if your goal is merely reporting on user activity *after* the fact (e.g., 'How many logins happened yesterday?'), you might need dedicated analytics tools that read logs; this MCP focuses on making changes and managing the core structure using tools like `list_admin_events` or `get_user`. The key difference: we manage the system's state, not just reading about it.

Managing user access rights is a manual nightmare.

Today, managing identity means clicking through endless dashboards. You have to open the realm list, drill down into client settings, then check groups for membership, and finally update roles in a separate section. If you need to audit just one user's permissions change from three months ago, it’s a multi-step process involving several screens and copy-pasting IDs.

With this MCP, the whole thing becomes conversational. You tell your agent what needs fixing—like 'Who updated the database client secret last week?'—and it instantly pulls that information using `list_admin_events`. The result isn't a spreadsheet; it’s an answer.

Keycloak MCP: Total control over your security infrastructure.

Manual tasks like deleting old client applications or resetting credentials are high-friction, multi-step processes. You're wasting time jumping between the user panel to the group manager just to complete a simple cleanup task.

The MCP brings that complexity into one chat window. It treats your entire security setup—users, clients, groups, realms—as programmable data, letting you execute `delete_client` or `create_role` in a single prompt.

Keycloak: 34 Tools for Identity Management

These tools give you granular control over every part of the Keycloak system, letting you manage users, clients, groups, and roles directly from your chat interface.

#	TOOL	DESCRIPTION
01	`create_auth_flow`	Builds a new authentication process flow within Keycloak.
02	`create_client`	Registers and creates a brand new client application in the realm.
03	`create_group`	Establishes a new, top-level user group for organization.
04	`create_role`	Defines and creates a new security role available at the realm level.
05	`create_user`	Creates an account for a brand new user in the system.
06	`delete_client`	Removes an existing client application from the realm entirely.
07	`delete_group`	Deletes a defined group, removing all associated users and roles.
08	`delete_realm`	Permanently deletes an entire security realm environment.
09	`delete_user`	Removes a user account from the system, making it permanently inactive.
10	`get_client_secret`	Retrieves the confidential secret key associated with a client application.
11	`get_client`	Fetches and displays all current details for a specified client.
12	`get_group`	Retrieves the full details of a specific user group.
13	`get_realm`	Fetches and displays all information for a specified security realm.
14	`get_role`	Retrieves the definition of a specific role by its name.
15	`get_user`	Fetches and displays all current details for a specific user account.
16	`import_realm`	Loads an entire realm environment into Keycloak from an external source.
17	`list_admin_events`	Retrieves a chronological list of all administrative changes made to a specific realm.
18	`list_auth_flows`	Lists all available authentication flows configured for the system.

#	TOOL	DESCRIPTION
19	list_client_roles	Displays all roles that can be assigned to a client application.
20	list_clients	Gets an overview of every client application registered in the realm.
21	list_groups	Displays the entire group hierarchy structure, showing parent-child relationships.
22	list_realms	Lists all accessible security realms managed by the instance.
23	list_required_actions	Identifies and lists actions that are required to proceed with certain changes.
24	list_roles	Displays all security roles available at the realm level for assignment.
25	list_user_groups	Lists which groups a specific user currently belongs to.
26	list_users	Retrieves a list of all active and inactive users within the specified realm.
27	logout_all_users	Forces every logged-in user to log out instantly across the entire realm.
28	partial_export_realm	Generates a partial data export of all settings and structures within a specific realm.
29	regenerate_client_secret	Creates a brand new secret key for an existing client application.
30	reset_user_password	Resets the password for a user without needing to know their previous credentials.
31	update_client	Modifies existing settings or metadata for a client application.
32	update_group	Changes the properties or membership of an established group.
33	update_realm	Modifies general settings and metadata for an entire realm environment.
34	update_user	Updates personal information or status details for a specific user account.

See It in Action

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

U List all realms available in our Keycloak instance.

I've retrieved the realms. You have 3 active realms: 'master', 'production-apps', and 'staging-environment'.

U Get the details for user ID '550e8400-e29b' in the 'production-apps' realm.

Fetching user data... User 'john.doe' found. Email: john@example.com, Status: Enabled, Created: 2023-10-12.

U Create a new group called 'Engineering-Leads' in the 'master' realm.

Group 'Engineering-Leads' has been successfully created in the 'master' realm.

Frequently Asked Questions

01 **How do I list all the environments available using Keycloak MCP?**

You use the `list_realms` tool. This command retrieves every active realm, letting you see exactly how many isolated security environments your instance manages.

02 **Can I reset a user's password with Keycloak MCP?**

Yes, you can use the `reset_user_password` tool. This lets you instantly reset any user's password without needing to know their current credentials or access the console.

03 **What is the difference between listing users and getting a user by ID using Keycloak MCP?**

The `list_users` tool provides an overview of all users in the realm. If you need specific, deep details about one person, you use the `get_user` tool with their unique identifier.

04 **How do I know who changed a setting last week using Keycloak MCP?**

You run `list_admin_events`. This tool gives you a comprehensive audit log, detailing administrative changes across the realm, including who made them and when.

05 **Does Keycloak MCP help me add new roles to users?**

Yes. After defining the role using `create_role`, you can update user memberships or group settings, which effectively applies that new role to the target user.

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 `"keycloak": { "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

Keycloak 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 Keycloak. 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	Keycloak MCP
Server ID	`019e38b4-e5ba-73cb-84be-54df298d68f3`
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/keycloak.