MCP SERVER | NO CODE | CLOUD HOSTED

Terraform Cloud (HCP) MCP

Orchestrate complex infrastructure changes via chat.

Terraform Cloud (HCP) allows your AI agent to manage your entire infrastructure lifecycle using natural language. You can list organizations, create projects, trigger runs, and extract specific state outputs directly from the cloud without opening a dashboard. It puts high-level governance controls and detailed run monitoring right into your chat window.

A+ Quality Score 98.33/100

infrastructure-as-code | provisioning | workspace-management | automation | cloud-ops | state-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/terraform-cloud-hcp

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.

Terraform Cloud (HCP) MCP

42 tools available
Cloud-hosted on Vinkius

Connecting your Terraform Cloud (HCP) account gives your agent direct control over your Infrastructure as Code (IaC) workflows. Instead of clicking through multiple dashboards, you can now talk to your environment. Your agent handles the complexity of the HCP API, letting you manage everything from high-level governance to minute resource changes. Need to see what changed? You can trigger a run and monitor its progress in real time. Want to enforce compliance? Use policies and variable sets to govern workspaces across organizations. When you connect this MCP via Vinkius, all your cloud environments become accessible through a single point of interaction with any MCP-compatible client.

Core Capabilities

01 — Manage Organization Structure

You can list, create, and delete entire organizations or projects to maintain high-level governance.

02 — Control Workspace Deployments

The agent can manage workspace locks, apply variable sets, and associate run tasks to specific workspaces for deployment control.

03 — Orchestrate Infrastructure Runs

Trigger new runs, plan changes, or discard incomplete plans directly through natural language commands.

04 — Extract State Data and Policies

Retrieve current state versions and pull specific output values to use in downstream analysis or automation scripts.

05 — Enforce Governance Rules

Create policies, set up variable sets, and manage user access controls across teams and organizations.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/terraform-cloud-hcp** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Terraform Cloud User or Team API Token.

02 Your AI client authenticates the connection, giving it visibility into your cloud environment structure.

03 You simply instruct your agent—for example, 'Plan an update for the production workspace'—and the tool executes the necessary sequence of calls.

The bottom line is you tell your agent what change needs to happen, and it handles all the complex API interactions required to make it real.

Built For

This tool solves the problem of context switching. If you're an operations engineer who spends half a day clicking between dashboards just to check resource state, this is for you. It turns complex cloud management into simple conversation.

DevOps Engineer

Automating routine workspace monitoring and running plans without ever leaving the chat interface.

Cloud Architect

Quickly inspecting state outputs and checking policy compliance across dozens of different organizations simultaneously.

SRE Team Lead

Troubleshooting failed runs, forcing unlocks on locked workspaces, and managing access controls during an incident response.

What Changes When You Connect

01 Manage the entire lifecycle without context switching. You can list organizations, create projects, and manage workspaces—all from your AI client's natural language prompt.

02 Gain full visibility into state management. Use `get_state_version_outputs` to pull specific output values, allowing you to use those results immediately in a subsequent step or script.

03 Control deployments precisely. You can run `create_run` and then `show_plan`, giving your agent the necessary details before committing changes with `apply_run`.

04 Enforce compliance automatically. Use tools like `create_policy` to define rules, making sure that every new deployment adheres to organizational standards before it goes live.

05 Handle incidents faster than ever. If a workspace is locked up, you don't need to navigate menus; simply ask the agent to execute `force_unlock_workspace` and get back to work.

Real-World Applications

Auditing Compliance Post-Deployment

A cloud architect needs proof that all staging environments use approved networking components. They ask their agent to `list_workspaces` in the 'Staging' organization, then run an `explorer_query`, and finally review the results against a set of defined policies.

Responding to a Broken Service

An SRE notices a critical workspace is locked. Instead of logging into the dashboard, they prompt their agent to execute `force_unlock_workspace`. Once unlocked, they can then run `create_run` and apply the fix.

Building Automated Pipelines

A DevOps engineer needs a new service. They ask their agent to first `create_organization`, next `create_project`, set up variables using `create_variable_set`, and finally, trigger the full deployment plan with `create_run`.

Extracting Secrets for Downstream Use

A platform engineer needs a specific ID from a newly deployed VPC. They ask their agent to get the current state version (`get_current_state_version`), retrieve the outputs (`get_state_version_outputs`), and feed that single value into another service's API call.

Patterns to Avoid

Trying to fix everything manually

✗ AVOID

A user tries to check the status of 20 workspaces, then navigates to each one individually to check its lock status and audit log history.

✓ INSTEAD

Use `list_workspaces` to see all targets at once. Then, if you need compliance data, run `list_audit_events`. For a full overview across the board, use `explorer_query`.

Running plans without approval

✗ AVOID

A user executes a plan and then immediately hits 'apply' in the dashboard without reviewing the output JSON first.

✓ INSTEAD

First, ask your agent to `show_plan`. Review the resulting change set. Once you confirm it looks right, tell the agent to execute `apply_run`.

Over-complicating governance setup

✗ AVOID

A user tries to manually assign team access by going through multiple menus and selecting individual users and resources one by one.

✓ INSTEAD

Use the agent's tools. First, `create_team` if needed. Then, use `add_team_workspace_access` to grant bulk permissions efficiently.

The Right Fit

You should use this MCP if your workflow involves managing resource state, governance policies, or complex deployment lifecycles within Terraform Cloud (HCP). If you need to create a new project, monitor run progress, apply variablesets, or enforce rules using `create_policy`, this is the right tool. Don't use it if your only goal is simple data retrieval that doesn't involve state management—for pure read-only tasks, a dedicated API connector might suffice. However, because of tools like `get_state_version_outputs` and `list_audit_events`, this MCP handles both reading the history and making controlled changes to the infrastructure itself.

The Cloud Dashboard Maze

Today, managing cloud resources means clicking through dozens of tabs. You open the organization view, then click into a project, find the specific workspace, and finally drill down to check run history or policy compliance. It’s slow, it requires context switching, and you're always worried about missing one crucial button.

With this MCP, your agent handles the clicks. Instead of navigating menus, you just talk to your environment. You can ask for all workspaces that need attention, or tell it exactly which run needs to be canceled. The result is a clean answer in your chat, not ten browser tabs open.

Control Infrastructure State with Terraform Cloud (HCP)

Previously, updating governance meant manually checking user permissions and recreating variable sets across different projects. If a policy changed, you had to remember which workspaces needed manual updates.

Now, you tell the agent to `create_policy` or run `apply_variable_set_to_workspace`. The system handles the rollout and validation automatically. You are defining rules at the top level, not patching things individually.

Terraform Cloud (HCP) with 36 Tools

These tools give your agent full control over provisioning, managing workspaces, enforcing policies, and auditing the entire infrastructure lifecycle within Terraform Cloud (HCP).

#	TOOL	DESCRIPTION
01	add_team_user	Adds a user to an existing team within your organization.
02	add_team_workspace_access	Grants specific teams access permissions for a particular workspace.
03	apply_run	Applies the planned changes from a run, committing them to your infrastructure.
04	apply_variable_set_to_workspace	Configures a workspace by applying an entire set of defined variables.
05	associate_run_task_to_workspace	Links a specific run task to a target workspace, ensuring proper execution flow.
06	cancel_run	Stops an active or pending infrastructure run immediately.
07	create_notification_configuration	Sets up alerts and notifications for changes happening within a workspace.
08	create_organization	Establishes an entirely new, top-level organizational boundary in your cloud account.
09	create_policy_set	Creates a group of governance policies that enforce specific architectural rules.
10	create_policy	Defines a single, reusable policy to check for compliance or mandate specific configurations.
11	create_project	Sets up a new container project under an existing organization structure.
12	create_registry_module	Creates a private, self-contained module that doesn't rely on version control system (VCS) integration.
13	create_registry_provider	Sets up and manages a private registry provider for resource management.
14	create_run_task	Creates a specific, repeatable task that must be executed during an infrastructure run.

#	TOOL	DESCRIPTION
15	create_run	Starts a new instance of an infrastructure run, which can generate plans or apply changes.
16	create_state_version	Saves the current state output as a distinct, historical version for record-keeping.
17	create_team	Creates a new team unit within your organization for role grouping and access control.
18	create_variable_set	Defines a collection of variables that can be consistently applied to multiple workspaces.
19	create_workspace	Initializes a new, dedicated workspace for deploying specific infrastructure components.
20	create_workspace_variable	Adds a single variable to an existing workspace, allowing custom input parameters.
21	destroy_organization	Permanently deletes an entire organizational structure and all associated resources.
22	discard_run	Aborts a run that is currently in progress or has not yet been finalized.
23	explorer_query	Runs a comprehensive query across multiple workspaces to find specific data points or tags.
24	force_unlock_workspace	Overrides the normal locking mechanism to make a workspace available for changes when needed.
25	get_current_state_version	Retrieves the metadata identifying the most recent state version of a given workspace.
26	get_plan_json	Fetches the detailed JSON output that represents exactly what changes will happen during an apply.
27	get_state_version_outputs	Retrieves all defined output values from a specific, historical state version of a workspace.
28	list_audit_events	Pulls a list of critical governance events that occurred within the organization.
29	list_organizations	Lists all top-level organizations you have access to in your account.
30	list_projects	Retrieves a list of all projects defined within a specific organization.

#	TOOL	DESCRIPTION
31	list_teams	Lists all established teams and their membership within an organization.
32	list_workspaces	Shows every workspace that exists inside a designated organization or project.
33	lock_workspace	Prevents any further changes to a workspace until the lock is explicitly released.
34	remove_team_user	Removes a specific user from membership in an existing team.
35	safe_delete_workspace	Initiates the process to safely and permanently delete a workspace, handling dependencies first.
36	show_apply	Displays detailed information about a past or planned apply action.
37	show_organization	Retrieves the full metadata and details for a specific organization.
38	show_plan	Shows detailed information about a generated plan, detailing resource changes before they happen.
39	unlock_workspace	Releases the lock on a workspace, allowing it to be modified again after an incident or run.
40	update_organization	Modifies existing details of an organization without recreating it.
41	update_team	Updates the name, description, or membership of a team unit.
42	upload_policy_code	Uploads custom code that will be evaluated by your governance policies.

See It in Action

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

U List all organizations I have access to in Terraform Cloud.

I've found 2 organizations: 'Acme-Corp' and 'Dev-Sandbox'. Which one would you like to explore?

U Show me all workspaces in the 'Acme-Corp' organization that have the tag 'production'.

Found 3 production workspaces: 'vpc-prod' (ws-K9...), 'eks-main' (ws-L2...), and 'rds-cluster' (ws-M5...).

U Get the output values for workspace ws-K9j2L8mP1.

The outputs for 'vpc-prod' are: vpc_id = 'vpc-0a1b2c', public_subnets = ['subnet-1', 'subnet-2'], and region = 'us-east-1'.

Frequently Asked Questions

01 **How do I check if a workspace is locked using the Terraform Cloud (HCP) MCP?**

You can use `list_workspaces` to see the current status. If you need to proceed despite the lock, your agent can execute `force_unlock_workspace`.

02 **Can I retrieve outputs from old state versions with Terraform Cloud (HCP) MCP?**

Yes. The tool `get_state_version_outputs` lets you pull specific output values from any historical state version, which is critical for auditing.

03 **Is this MCP safe to use when running destructive commands like destroy on Terraform Cloud (HCP)?**

The agent guides the process. Before destruction, you should always use `show_plan` to review exactly what resources will be removed before executing a command.

04 **How does the Terraform Cloud (HCP) MCP handle user access?**

You manage access using tools like `add_team_user`, `remove_team_user`, and `add_team_workspace_access` to maintain strict role-based governance.

05 **What if I need to update a team name? Can the Terraform Cloud (HCP) MCP do that?**

Yes, you can modify existing team details using the `update_team` tool. This keeps your organization's structure current without manual intervention.

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 `"terraform-cloud-hcp": { "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

Terraform Cloud (HCP) 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 Terraform Cloud (HCP). 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	Terraform Cloud (HCP) MCP
Server ID	`019e38f9-47e8-717c-9b80-d56cf37b5fe6`
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/terraform-cloud-hcp.