MCP SERVER | NO CODE | CLOUD HOSTED

NetBird MCP

Manage your entire mesh network from chat.

NetBird controls your Zero Trust mesh network directly through conversation. Use this MCP to manage accounts, create users, define access policies, and configure resources without logging into a dashboard. It gives your AI agent full control over identity and network infrastructure.

B Quality Score 81.32/100

zero-trust | vpn | network-management | access-control | mesh-network

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

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.

NetBird MCP

89 tools available
Cloud-hosted on Vinkius

You don't have to navigate multiple dashboards just to onboard one user or update a single policy. Connect your NetBird account through Vinkius to give your AI client direct administrative access to your entire private network. Your agent acts like an experienced net admin, handling complex workflows via natural language prompts.

Need to add a new subnet? You can ask for it and get the resource created instantly. Want to adjust who can talk to whom? Simply tell your agent which policies need updating. If you're managing user lifecycle—like creating service accounts or revoking access—it handles all the necessary steps, from generating invite links to changing passwords. This setup means that whether you’re working in an IDE, a terminal, or through any other AI client, you maintain full control over your network perimeter without leaving your workflow.

Core Capabilities

01 — Manage user identities and accounts

Using tools like list_users and create_user, your agent can track all users, invite new ones, or set up service accounts.

02 — Control network access policies

Your agent lets you define rules for traffic using tools like create_policy, ensuring only authorized services can communicate across the mesh network.

03 — Configure network resources

You can provision new infrastructure components by listing or creating networks, routers, and specific host subnets with tools like list_networks and create_network_resource.

04 — Handle user onboarding and offboarding

Manage the full life cycle using functions that generate user invite links (create_user_invite), approve pending accounts (approve_user), or delete users entirely (delete_user).

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and enter your NetBird API Token.

02 Second, send a natural language request to your AI client, specifying the action you need (e.g., 'List all users who haven't logged in').

03 Finally, your agent executes the necessary tool calls, retrieves the data, and presents it back to you for review or confirmation.

The bottom line is that your AI client treats your network infrastructure like a backend API, letting you manage everything through conversation.

Built For

This MCP is built for system administrators and engineers who deal with complex, constantly changing network environments. It targets the pain of context switching: having to jump from their terminal to a web dashboard just to perform routine access control tasks.

DevOps Engineer

They use this MCP to automate infrastructure changes, such as creating new networks or updating policies, directly within their primary coding environment.

IT Administrator

They rely on it for user lifecycle management. Instead of clicking through multiple tabs, they ask the agent to generate invites and manage access approvals in one go.

Security Analyst

The analyst uses this MCP to audit current network state by listing all active peers or checking logs for suspicious activity immediately.

What Changes When You Connect

01 Stop logging into dashboards. Instead of navigating multiple UI sections to update a user's role or block status, you simply ask your agent to use the update_user tool and get it done instantly.

02 Audit trail control: You no longer have to manually compile audit reports from different tabs. Simply ask your agent to list_audit_events and get a comprehensive log of every activity that happened.

03 Efficient user onboarding: Instead of creating an account, then sending a link, you can use create_user_invite followed by resend_user_invite to manage the entire process without leaving your chat window.

04 Network resilience: If a peer's IP changes or their SSH status needs updating, calling update_peer ensures that configuration is consistent across all records. It keeps your network running smoothly.

05 Security hardening: You can enforce strict security protocols by using create_policy to define precise access rules between sources and destinations, going far beyond simple firewall rules.

Real-World Applications

Offboarding a former employee

The Security Analyst needs to cut off all access for an ex-employee immediately. They prompt the agent: 'Disable user Bob and delete his credentials.' The agent executes delete_user, updates their profile using update_user (to ensure group removal), and finally deletes any associated tokens via delete_user_token.

Adding a new service subnet

The DevOps Engineer needs to connect a newly provisioned server rack. They ask the agent to 'Add the 10.2.3.0/24 subnet and create a network resource.' The agent uses create_network_resource, then updates the relevant policy using update_policy so traffic is allowed.

Handling pending user access requests

The IT Administrator receives an invite for a contractor. Instead of logging into the admin panel, they tell the agent to 'Approve the request from Jane Doe.' The agent runs approve_user and confirms the status change.

Reviewing network connectivity issues

The Security Analyst suspects a rogue connection. They ask the agent to list all connected peers, running list_peers to identify the suspicious MAC address, then use get_peer to check its last reported IP and location.

Patterns to Avoid

Manual manual provisioning

✗ AVOID

The user logs into the dashboard, manually navigates to 'Users,' creates a new account for Jane, then copies an invite link, sends it via email, and finally checks the status page later.

✓ INSTEAD

Instead, ask your agent to run create_user_invite. Then, if they need to resend it, just call resend_user_invite. The entire process stays within your chat window.

Forgetting necessary cleanup steps

✗ AVOID

The user deletes a network resource using delete_network_resource but forgets to update the associated security policy rules.

✓ INSTEAD

Always verify policies first. Use get_policy to check current rules, and then use update_policy before calling delete_network_resource. This ensures no access is accidentally broken.

Guessing user roles

✗ AVOID

The user tries to change a group's membership directly via chat without knowing if the user account needs an explicit role update first.

✓ INSTEAD

Always check the current state. Run get_user before attempting updates, and use update_user to guarantee both the group assignment and the core user profile are updated correctly.

The Right Fit

Use this MCP if your job involves managing complex network infrastructure where identity, resources, and policies interact constantly. You need automated workflows for things like onboarding users (create_user), adjusting access rules (update_policy), or provisioning new hardware (create_network_resource). This is a pure network operations tool.

Don't use it if you just need to list static data that doesn't require action, such as simply viewing country codes. For basic lookup tasks, an API catalog might be better. Also, don't use this for non-network related administrative tasks like managing billing or user payroll; stick to access control and topology management.

The Pain of Dashboard Hopscotch

Today, updating your network setup feels like playing a game of digital hopscotch. You have to jump between the User Management dashboard, then click over to the Policies tab, and finally go back to the Network Resources section just to ensure everything connects correctly. Every single change means logging in, clicking through menus, manually confirming details, and copying/pasting IDs.

With this MCP, all of that manual hopping disappears. Your AI client handles the entire workflow when you talk to it. You state your goal—like 'Add a new subnet and allow traffic from Group A'—and the agent executes multiple steps in sequence, giving you only the final confirmation.

NetBird MCP: Full Network Control

You no longer have to manually run commands or click through five different tabs just to onboard a new user. The agent handles generating the necessary invite link using create_user_invite, managing their status with approve_user, and updating their roles instantly via update_user.

It's simple: your conversation becomes your command line. You manage every aspect of your Zero Trust network from one place.

NetBird MCP with 89 Tools

These tools let you programmatically perform every administrative action needed to configure, monitor, or change your entire network topology.

#	TOOL	DESCRIPTION
01	accept_user_invite	Allows an unauthenticated user to accept an invite link and set a password.
02	approve_user	Approves the status of a pending user account.
03	change_user_password	Resets or changes a user's password through an embedded identity provider.
04	create_group	Creates a new logical group for organizing users and resources.
05	create_msp_tenant	Sets up a brand new managed service provider (MSP) tenant account.
06	create_nameserver	Creates a dedicated group for nameservers within the network structure.
07	create_network_resource	Adds an actual resource, such as a host subnet or domain name, to a specific network.
08	create_network_router	Deploys a new router device within the overall network topology.
09	create_network	Establishes an entirely new, isolated virtual network environment.
10	create_policy	Defines a granular rule set specifying what protocols and ports are allowed between sources and destinations.
11	create_posture_check	Sets up a security check that verifies parameters like OS version or geographic location for connecting peers.
12	create_route	Creates an outdated network route entry.
13	create_setup_key	Generates a setup key, which can be used for one-time or reusable authentication access.
14	create_temporary_access_peer	Creates temporary credentials allowing limited network access to a specific peer.

#	TOOL	DESCRIPTION
15	create_user_invite	Generates a unique link that can be shared with a new user to join the system.
16	create_user_token	Creates a new, secure personal access token for long-term use by an individual user.
17	create_user	Registers either a service account or invites a standard human user into the system.
18	delete_account	Removes an entire NetBird account and all associated resources permanently.
19	delete_group	Deletes a defined logical group from the network structure.
20	delete_nameserver	Removes an entire nameserver grouping from service.
21	delete_network_resource	Deletes a specific host, subnet, or domain resource.
22	delete_network_router	Removes a router device from the network topology.
23	delete_network	Deletes an entire virtual network environment, removing all contained resources.
24	delete_peer	Delete a peer
25	delete_policy	Removes a specific access policy rule set.
26	delete_posture_check	Takes down an existing security posture verification check.
27	delete_route	Removes an outdated network route entry.
28	delete_setup_key	Invalidates and removes a previously generated setup key.
29	delete_user_invite	Removes an existing, unsent user invitation link.
30	delete_user_token	Revokes and deletes a personal access token for a specific user.
31	delete_user	Permanently removes a registered user from the network system.
32	get_current_user	Retrieves detailed information about the account currently authenticated and using the service.
33	get_dns_settings	Fetches global settings related to DNS management for the network.
34	get_group	Retrieves detailed information about a specified group.

#	TOOL	DESCRIPTION
35	`get_nameserver`	Fetches all details regarding a specific nameserver grouping.
36	`get_network_resource`	Retrieves the current configuration and status of a specified network resource.
37	`get_network_router`	Fetches detailed operational data for a specific router device.
38	`get_network`	Retrieves the overall configuration and status of a designated network.
39	`get_peer`	Fetches detailed information about an individual connected peer device or user.
40	`get_policy`	Retrieves the full details of a specific access policy rule set.
41	`get_posture_check`	Fetches the current rules and status of a security posture verification check.
42	`get_public_user_invite`	Retrieves details about an invitation link that is publicly accessible without logging in.
43	`get_route`	Retrieves outdated network route information.
44	`get_setup_key`	Fetches the details of a specific setup key, confirming its status and use.
45	`get_user_token`	Retrieves information about a specific user access token.
46	`invite_msp_tenant`	Sends an invitation to an existing account, turning it into a managed service provider tenant.
47	`list_accessible_peers`	Lists all the network peers that are reachable from the current peer's location.
48	`list_accounts`	Retrieves a list of every NetBird account connected to the system.
49	`list_all_network_routers`	Lists all routers deployed across every network in the environment.
50	`list_audit_events`	Retrieves a chronological list of all system audit events, showing who did what and where.
51	`list_cities`	Returns a list of city names associated with a given country code.
52	`list_countries`	Provides all available ISO 3166-1 alpha-2 two-letter country codes.

#	TOOL	DESCRIPTION
53	list_groups	Lists every defined logical group within the network structure.
54	list_msp_tenants	Retrieves a list of all managed service provider tenants under the umbrella account.
55	list_nameservers	Lists every dedicated nameserver group configured in the network.
56	list_network_resources	Retrieves a list of all specific resources (subnets, domains) within a given network.
57	list_network_routers	Lists every router device deployed in a specified network.
58	list_network_traffic_events	Retrieves experimental data showing network traffic events for analysis.
59	list_networks	Provides a list of every virtual network configured in the system.
60	list_peers	Lists all registered and active peers connected to the mesh network.
61	list_policies	Retrieves a list of every defined access policy rule set.
62	list_posture_checks	Lists all currently active security posture checks on the network.
63	list_proxy_events	Retrieves logs detailing access attempts through the reverse proxy layer.
64	list_routes	Lists outdated network route entries.
65	list_setup_keys	Shows a list of all setup keys currently active in the system.
66	list_user_invites	Retrieves a queue of pending user invitation links awaiting action.
67	list_user_tokens	Lists all personal access tokens associated with a specific user account.
68	list_users	Retrieves the full list of every registered user in the system.
69	regenerate_user_invite	Creates a fresh, new invite link for an existing user account.
70	reject_user	Marks a pending user invitation as rejected without sending the notification.
71	resend_user_invite	Resends an existing, previously generated user invitation link to the intended recipient.

#	TOOL	DESCRIPTION
72	respond_msp_tenant_invite	Accepts or declines a formal Managed Service Provider (MSP) tenant invitation.
73	unlink_msp_tenant	Separates an MSP tenant account from its current owner to assign it to someone else.
74	update_account	Modifies global account settings, such as changing peer login expiration dates or network ranges.
75	update_dns_settings	Adjusts the system's DNS parameters, for example, enabling or disabling management groups.
76	update_group	Modifies a group by changing its name, adding peers, or assigning new resources to it.
77	update_msp_tenant_subscription	Creates or updates the subscription details for an existing MSP tenant.
78	update_msp_tenant	Changes the name of a tenant or modifies its associated access groups.
79	update_nameserver	Updates the configuration details for an existing nameserver group.
80	update_network_resource	Modifies a specific network resource, such as changing its IP range or domain name.
81	update_network_router	Updates the operational status or configuration of an existing router device.
82	update_network	Changes the descriptive name and details of a virtual network environment.
83	update_peer	Modifies peer information, such as changing its SSH status or internal IP address.
84	update_policy	Changes the rules governing an existing access policy rule set.
85	update_posture_check	Modifies the criteria used in a security posture check, like requiring a specific OS version.
86	update_route	Updates outdated network route information.
87	update_setup_key	Modifies an existing setup key, allowing you to revoke it or change its associated groups.
88	update_user	Updates a user's role, assigned resource groups, or block status without deleting the account.

#	TOOL	DESCRIPTION
89	verify_msp_tenant_dns	Validates that the domain name DNS challenge for an MSP tenant is correctly configured.

See It in Action

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

U List all users in my NetBird account.

I've retrieved the user list. You have 3 active users: Admin (ID: user_1), Developer A (ID: user_2), and Service_Bot (ID: user_3). Would you like to manage any of them?

U Create a new user invite link.

I've generated a new invite link (ID: invite_abc123). You can share this with the new user to join your network.

U Show me the details of the current authenticated user.

The current user is 'Admin' with email 'admin@company.com'. Your role is 'Owner' and you have full access to the account settings.

Frequently Asked Questions

01 **How do I change a user's role using NetBird MCP?**

You update the user's profile directly by calling update_user. This function allows you to manage their assigned roles, auto-groups, or block status without deleting their account.

02 **What if I need to create a whole new isolated network?**

You start by using the create_network tool. Once the container is established, you can then add resources and routers inside it using tools like create_network_resource or create_network_router.

03 **Can I see who has accessed my network recently? (list_audit_events)**

Yes. Use list_audit_events to retrieve a full, chronological list of every system activity that occurred across your NetBird account, showing the initiator and target.

04 **How do I revoke someone's access credentials?**

To completely remove credentials, you can use delete_user or delete_user_token. If you just need to stop them from logging in, update_user lets you change their block status.

05 **What is the difference between create_user and create_user_invite?**

create_user establishes an account (often for a service user), while create_user_invite generates a single, temporary link used to onboard a human 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 `"netbird": { "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

NetBird 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 NetBird. 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	NetBird MCP
Server ID	`019e38c6-fd4c-727d-829e-2218d7b7b685`
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/netbird.