MCP SERVER | NO CODE | CLOUD HOSTED

Buildkite MCP for AI Agents

Manage CI/CD Pipelines and Build Agent Status

Buildkite connects your CI/CD pipelines to any AI agent, letting you manage complex build workflows using natural conversation. You can instantly list pipelines, trigger new tests on specific branches, and deep-dive into logs—all without leaving your chat interface.

A+ Quality Score 100/100

ci-cd | pipeline-automation | deployment | build-management | devops

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

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.

Buildkite MCP

11 tools available
Cloud-hosted on Vinkius

Stop context-switching between the terminal and web consoles just to check a build status. This MCP lets your AI agent handle full CI/CD lifecycle management through conversation. You can ask it to list all pipelines across the organization, trigger an ad-hoc test run on a specific feature branch, or instantly cancel a stuck deployment. It also handles everything in between: getting deep details about past runs and verifying which build agents are active.

It’s like giving your agent full control over your entire build operation center. Whether you're running local tests or coordinating hybrid infrastructure, the platform makes it simple to monitor builds for an entire company. You just connect this Buildkite MCP via Vinkius and start managing deployments conversationally.

Core Capabilities

01 — List all organizational pipelines

Get a complete overview of every active build pipeline configured across your organization.

02 — Trigger ad-hoc builds

Start a new, immediate test run for any specific pipeline or branch.

03 — Cancel and restart deployments

Halt running builds that got stuck or quickly initiate a rebuild of a failed process.

04 — Monitor build agents globally

Verify the status of all connected build agents to ensure your infrastructure is online.

05 — Retrieve detailed build history

Fetch deep logs and metadata for any past pipeline execution, helping pinpoint failure causes.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and provide your Buildkite API Token and Organization Slug.

02 Your AI agent authenticates the connection and confirms access across your organization's build environment.

03 You use natural language prompts (e.g., 'What failed builds did we have yesterday?') and your agent executes the necessary commands to deliver actionable data.

The bottom line is, you treat managing complex CI/CD infrastructure like talking to a teammate who already knows where all the buttons are.

Built For

This MCP targets anyone responsible for keeping software moving. It's perfect for the DevOps engineer tired of constant dashboard refreshing, or the tech lead who needs fast build failure summaries before a major merge.

DevOps Engineer

Uses this MCP to orchestrate hybrid CI infrastructure, monitor hanging processes, and cancel stuck builds effortlessly without switching tools.

Software Developer

Triggers ad-hoc test runs on specific feature branches right from their IDE when they need fast feedback before committing code.

Tech Lead

Gets a clean, summarized view of the team's overall build failure rates across multiple pipelines to guide merging decisions.

What Changes When You Connect

01 Instantly cancel stuck builds or retry failures. You can use the `cancel_build` tool to stop a running job immediately, saving compute time and keeping deployments moving.

02 Deep visibility into historical data. By calling `list_all_builds` and then `get_build`, you pinpoint exactly when and why a failure occurred months ago.

03 Global agent monitoring is simple. Use `list_agents` to verify that all your distributed build agents are online, which is crucial for hybrid infrastructure.

04 Never lose track of pipelines again. The ability to use `list_pipelines` gives you an instant inventory of every defined workflow in the company.

05 Speed up iteration cycles. Need a quick test? Use `create_build` to trigger new runs on feature branches without leaving your chat window.

Real-World Applications

The staging deployment is failing, and I need to know why.

Instead of checking the console logs for hours, you ask your agent. It uses `list_pipeline_builds` to find the latest failed runs, then calls `get_build` on that specific build to summarize the job failures and point you exactly where to look.

We have a critical bug found in production; I need an immediate rollback test.

You prompt your agent. It identifies the correct pipeline, uses `create_build` to trigger a fresh build on the stable branch, and monitors the results until it confirms readiness for deployment.

I suspect one of our remote build agents is offline.

You ask your agent about infrastructure health. It executes `list_agents`, showing you a real-time list of every registered agent and their operational status, letting you know immediately if parts of the team are disconnected.

The main branch build keeps failing right after merge.

You ask your agent to review the situation. It uses `list_pipelines` to find the source pipeline, and then it can use `rebuild` on a specific failed execution ID to rule out transient network issues.

Patterns to Avoid

Manually checking build history

✗ AVOID

The user logs into the Buildkite UI, navigates through multiple dropdown menus, and clicks on individual builds one by one to track a bug.

✓ INSTEAD

Ask your agent to use `list_all_builds` or `list_pipeline_builds`. This aggregates the necessary history data instantly, giving you an overview without clicking anything.

Forgetting which pipelines exist

✗ AVOID

A developer needs a build for the new microservice but doesn't know if it was registered as 'auth-api' or 'user-svc'. They waste time searching documentation.

✓ INSTEAD

Just ask your agent to use `list_pipelines`. It gives you an immediate, comprehensive inventory of every available pipeline name.

Assuming a failed build is fixed

✗ AVOID

A team member sees a failure and manually pushes code believing the issue was external. They don't run a proper verification test.

✓ INSTEAD

Instead, ask your agent to use `create_build` on that pipeline immediately. This forces a clean, verifiable execution of all current code against the latest rules.

The Right Fit

Use this MCP if you need conversational control over complex build processes—like triggering tests or getting historical logs without leaving your IDE. It's for teams whose daily work involves constantly checking build statuses across multiple pipelines, and needing to manage agents as part of the overall deployment picture. Don't use it if your only goal is simple API key management; you can handle that with basic token verification tools. If you just need a single source of truth on *which* builds ran, stick to `list_all_builds`. But if you need to diagnose or act on those builds (like canceling them), this MCP is what you need.

Buildkite MCP for AI Agents: Diagnosing CI/CD Build Failures

Right now, diagnosing a build failure means context-switching. You're in your IDE, see the red X, and then you have to open a web browser, log into Buildkite, find the right pipeline, locate the specific failing run, click through multiple tabs, and finally copy/paste the error details back into your chat.

With this MCP, you simply ask your agent about the failure. It uses `get_build` or `list_pipeline_builds`, pulls the relevant logs instantly, and summarizes the root cause—all in plain text right where you are working.

Buildkite MCP for AI Agents: Monitoring Distributed Build Agent Health

Before, checking infrastructure health was a manual chore. You'd have to remember which teams run on specific agents and then check dashboards one by one to make sure nothing had gone down in different regions or environments.

Now, you just ask your agent about the build agents. It executes `list_agents`, giving you an immediate report card of every single machine running your code. You know instantly if you have coverage gaps before a deployment even starts.

Buildkite MCP: 11 Tools for Pipeline Automation

Use these specific tools to list, trigger, cancel, or inspect every aspect of your CI/CD pipeline flow using conversational prompts.

#	TOOL	DESCRIPTION
01	`cancel_build`	Stops an actively running software build process immediately.
02	`create_build`	Initiates a new build execution for any configured pipeline.
03	`get_access_token_info`	Retrieves basic information confirming the status and scope of the currently connected API token.
04	`get_build`	Fetches all detailed metadata about a single, specific build run.
05	`get_pipeline`	Retrieves the full configuration and details for one named pipeline.
06	`list_agents`	Pings and lists all operational build agents connected to your organization.
07	`list_all_builds`	Retrieves a list of every build run that has occurred across the entire company setup.
08	`list_organizations`	Lists all separate Buildkite organizations the connected API token can access.
09	`list_pipeline_builds`	Gets a list of recent build runs associated with one specific pipeline.
10	`list_pipelines`	Provides an inventory of all pipelines configured within the current organization.
11	`rebuild`	Forces a re-execution of a specific build, useful if initial results were inconclusive.

See It in Action

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

U What were the last three builds that ran on our main production pipeline?

Build History Summary

- ✅ Build #204: **Status:** Passed (Main Branch)

Duration:* 5m 12s

Commit:* `a1b3c9d`

- ❌ Build #205: **Status:** Failed (Frontend Refactor)

Failure Point:* Unit Tests in `src/components/UI.js`.

Action:* Needs review of log segment 4.2.

- ✅ Build #203: **Status:** Passed (Main Branch)

Duration:* 5m 10s

*(Would you like me to inspect the failure logs for #205?)

U Trigger a fresh test run on the user-auth service with the latest code.

Build Initiated Successfully

The `user-auth` pipeline is starting a new build. You can track its progress here: [Live Link]

- **Target:** Latest Code (HEAD)
- **Pipeline ID:** 456
- **Status:** Running (Agent pinged successfully)

Please monitor this link; I'll let you know if the status changes.

U Can we check which build agents are currently online?

Here is the current operational status of your connected build agents:

AGENT NAME	STATUS	LAST PING TIME
US-East-1-Runner	✅ Online	2 minutes ago
EU-West-2-Worker	✅ Online	3 minutes ago
Asia-Pacific-Test	⚠️ Offline	Unknown

The agent in the Asia Pacific region appears to be disconnected. Should I run a ping test?

Frequently Asked Questions

01 **Can my AI agent restart failed builds for a specific branch?**

Yes. Ask the agent to find failed builds across your pipeline by using the list builds tool. Once it locates the specific build number, it can run the rebuild tool instantly, eliminating the need to search through hundreds of logs on the dashboard.

02 **How can I check the status of my physical runner agents?**

Ask your agent to list all agents connected to the Buildkite organization. It returns their UUIDs, operating systems, and connection state. If a runner hangs offline, your AI can immediately flag it to the Platform team, saving crucial deployment time.

03 **If a commit is pushed to 'main', can the agent trigger a fresh pipeline deployment?**

Absolutely. You can provide the commit SHA (or simply ask it to target 'HEAD' on the 'main' branch) and ask the agent to create a new build. It will hit the Buildkite trigger endpoint with a message of your choosing.

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 `"buildkite": { "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

Buildkite 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 Buildkite. 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	Buildkite MCP
Server ID	`019d7565-e350-724d-ace5-f6dac6a40203`
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/buildkite.