MCP SERVER | NO CODE | CLOUD HOSTED

Katalon TestOps MCP

Automate Quality Checks with Conversation.

Katalon TestOps (AI Test Management) lets you manage your entire quality pipeline using natural conversation. Connect your existing Katalon TestOps account to any AI client, and your agent handles everything from listing test runs to auditing complex software releases, letting you get immediate visibility into build health without clicking through dashboards.

A+ Quality Score 100/100

test-orchestration | quality-assurance | test-automation | software-testing | ci-cd-pipeline | test-analytics

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/katalon-testops-ai-test-management

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.

Katalon TestOps (AI Test Management) MCP

10 tools available
Cloud-hosted on Vinkius

You can treat your quality assurance process like a conversation with an expert teammate. This MCP connects directly to your Katalon TestOps account, giving your AI client full control over test management and quality orchestration. Instead of jumping between tabs or writing complex scripts, you just talk to your agent. It handles listing all available projects so you know exactly what's being tested, and it lets your agent trigger immediate re-executions of specific suites when a fix is ready for verification. Need deep details? Your agent pulls detailed test outcome summaries, including pass/fail rates and error stack traces. You can even audit the entire deployment process by listing configured execution environments or tracking defined software releases. If you're used to finding these connections piecemeal, Vinkius makes it simple: connect once from any MCP-compatible client and gain access to all your critical test data in one place.

Core Capabilities

01 — Orchestrate Test Runs

Tell the agent which tests failed and have it automatically trigger a re-run of that specific suite.

02 — Audit Software Releases

Get an overview of defined software releases, tracking aggregated test run statistics to verify build quality before deployment.

03 — Inspect Test Results

Retrieve detailed pass/fail rates, execution durations, and even extract application logs or visual screenshots from individual tests.

04 — Track Project Status

List all projects within your Katalon account to understand team assignments and suite counts across multiple repositories.

05 — Manage Build History

Review defined software builds, seeing their unique identifiers and how many associated test runs they contain.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/katalon-testops-ai-test-management** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Katalon Email and API Key credentials.

02 Connect the MCP to your preferred AI client (like Claude or Cursor).

03 Ask your agent questions like, 'What are the latest 5 test runs for project X?' and receive instant, structured data.

The bottom line is you get full visibility into your test suite's health by simply talking to your AI client instead of navigating complex web dashboards.

Built For

This MCP is for QA Managers who need a single source of truth on release readiness, or Automation Engineers tired of manually checking logs across different builds. If you spend time jumping between build pipelines and test reports, this is for you.

QA Manager

Monitoring the overall health of a software release; they use this to track team testing progress and check if all necessary quality gates are met before launch.

Automation Engineer

Debugging failures by running specific failing test suites immediately, then inspecting diagnostic logs directly from their workspace for rapid iteration.

DevOps Engineer

Auditing execution environments and verifying build quality results to ensure the entire CI/CD pipeline maintains its integrity across different operating systems and browsers.

What Changes When You Connect

01 You save time by getting immediate status updates. Instead of manually checking the dashboard, you can ask your agent to list test runs and instantly see which ones passed or failed.

02 Debugging gets faster. If a run fails, your agent doesn't just give you 'Fail.' It pulls detailed test results so you get error messages and stack traces right away.

03 You maintain full visibility over deployments. You can check the status of defined software releases using list_project_releases to ensure the build is ready before going live.

04 Never forget which environment a bug appeared in. By listing execution environments, you can confirm if the issue only happens on Safari or only on Windows 10.

05 Rerunning tests becomes instant. Instead of clicking through menus to re-trigger a failed test, your agent handles it using rerun_test_run and gives you the new session ID.

Real-World Applications

Investigating an intermittent failure

A QA Engineer notices a bug only appears on specific devices. They ask their agent to list execution environments, confirming if the issue is confined to Android 12 or if it's widespread across multiple target OS settings.

Checking release readiness quickly

A QA Manager needs a quick sign-off. They prompt their agent to list project releases, instantly seeing that the 'Q3 Beta' build has 95% pass rates but also flagging two specific critical tests that failed.

Auditing CI/CD pipeline integrity

A DevOps Engineer wants to verify if a recent deployment impacted stable builds. They use list_project_builds to see the last ten builds and then ask the agent to get details on the most recent one, checking its associated test runs.

Verifying a hotfix

The development team pushes an urgent fix. The engineer uses list_test_runs to identify the failing run ID and then tells the agent to rerun_test_run, immediately getting a new session ID for monitoring.

Patterns to Avoid

Treating it like basic status reporting

✗ AVOID

Manually running a query to simply check if a project exists or what the overall pass rate is. This requires simple data retrieval, not complex orchestration.

✓ INSTEAD

If you only need to know which projects are available, use list_projects. If you need deeper analysis (like re-running tests or auditing releases), this MCP is necessary.

Attempting manual environment tracking

✗ AVOID

Trying to remember if a bug was seen on Chrome/MacBook Pro vs Edge/Windows because documentation spread across multiple sheets.

✓ INSTEAD

Use list_execution_environments. This tool centralizes all target OS, browser, and device information in one place for easy comparison.

Forgetting to track release scope

✗ AVOID

Deploying a fix without knowing which official software release it belongs to, creating deployment confusion.

✓ INSTEAD

Always use list_project_releases first. This tool tracks defined releases and links them to aggregated test run statistics, keeping your quality reports accurate.

The Right Fit

Use this MCP if your primary pain point is coordinating complex, multi-stage testing processes—specifically, when you need visibility into build history, release tracking, and the ability to re-trigger tests based on live feedback. You should use it whenever a developer or QA needs to confirm that a fix works across multiple environments (like using list_execution_environments) or verify if a deployment is officially tied to a specific version (using list_project_releases). Don't use this MCP if you simply need to store test data; for raw data storage and basic reporting, a dedicated database connection might be simpler. Also, if your goal is just to pull metadata about projects without any testing capability, the list_projects tool handles that perfectly, but you won't get the full orchestration power.

The quality assurance workflow used to involve too much clicking.

Today, when a test fails, your process involves navigating deep into the Katalon TestOps dashboard. You check `list_test_runs` for the failure ID, then click into that run to see individual results, and finally copy-paste error stack traces or screenshots into a ticketing system. This cycle of clicking through tabs just to get one piece of information is exhausting.

With this MCP, you simply ask your agent: 'What happened in test run 986?' Your agent handles the entire journey—it pulls the detailed results, extracts the error messages, and presents them all back to you. You don't click a thing; you just talk.

---

get_test_result: Instant diagnosis for every failure.

Before this MCP, getting the full context of a single failed test meant checking dozens of fields—the duration, the status, the specific error message, and sometimes even needing to find a related application log. It was detective work, wasting critical time.

Now, when your agent uses get_test_result, you get all that information condensed into one actionable block of text. You know exactly what went wrong, why it failed, and where the error originated—all without leaving your chat window.

Katalon TestOps (AI Test Management) with 10 Tools

These tools let your AI agent interact with every part of your Katalon TestOps account, from listing projects to re-running specific failed test suites.

#	TOOL	DESCRIPTION
01	`list_projects`	Retrieves a list of all configured projects, including their names, IDs, and descriptions on Katalon TestOps.
02	`get_project`	Fetches full details about a specific Katalon project by its ID or name.
03	`list_test_runs`	Lists all test runs for a given project, showing their IDs, pass/fail status, total counts, and duration.
04	`get_test_run`	Retrieves the complete details for one specific Katalon test run.
05	`rerun_test_run`	Initiates a fresh execution of an existing test run, returning the ID of the new active session.
06	`list_test_results`	Lists detailed results for a specific test run, showing statuses and error messages.
07	`get_test_result`	Gets comprehensive information about a single recorded test result instance.
08	`list_execution_environments`	Lists all configured operating system, browser, and device environments available for testing within a project.
09	`list_project_releases`	Returns information about defined software releases, including their scheduled dates and aggregate test run metrics.
10	`list_project_builds`	Lists all associated software builds within a project, providing identifiers and the total count of linked test runs.

See It in Action

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

U List the last 5 test runs in project 'E-commerce API' (ID: 123)

I've retrieved the latest 5 test runs for project 123. Highlights: Run #987 (Passed), Run #986 (Failed - 2 errors), Run #985 (Passed). Would you like to inspect the results for the failed run?

U Rerun test run ID 986

Rerunning test run 986... New active execution started. New Run ID: 988. I'll let you know once the results are available. Would you like me to notify you if it fails again?

U What are the active releases defined in project 123?

I've identified 2 active releases in project 123: 'Q1 Spring Update' (v1.2.0) and 'Hotfix Security' (v1.2.1). Release v1.2.0 has 45 associated test runs with a 98% pass rate. Would you like a detailed build report?

Frequently Asked Questions

01 **How do I find out what projects are available using Katalon TestOps (AI Test Management) MCP?**

You use list_projects. This tool immediately retrieves a complete list of all projects associated with your account, giving you the names and IDs needed to focus your queries.

02 **Can I force a test run to re-execute after a fix?**

Yes, use rerun_test_run. This tool takes an existing run ID and kicks off a brand new execution, providing you with the ID for the fresh session so you can monitor it.

03 **What is the difference between list_project_builds and list_project_releases?**

list_project_builds shows granular software versions (the 'how' of the build), while list_project_releases tracks major, defined product releases that contain multiple builds. Use both to get a full view.

04 **Does Katalon TestOps (AI Test Management) MCP show me which browsers were used?**

Yes. You can use list_execution_environments to see all configured testing environments, including the specific OS, browser types, and device distributions you've covered.

05 **How do I check if a build passed quality checks?**

You first use list_project_releases to find the target release. Then, your agent can pull aggregated stats from that release, showing overall pass/fail rates and associated test runs.

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 `"katalon-testops-ai-test-management": { "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

Katalon TestOps (AI Test Management) 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 Katalon TestOps (AI Test Management). 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	Katalon TestOps (AI Test Management) MCP
Server ID	`019d75bf-bd01-718b-ab20-7a3822d41c2f`
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/katalon-testops-ai-test-management.