MCP SERVER | NO CODE | CLOUD HOSTED

Jira Software Cloud MCP

Automate boards, sprints, and issue movement.

Jira Software Cloud MCP connects your agent directly to your Jira instance, letting you manage Agile workflows without ever opening the browser. You can list boards, track sprints, inspect backlogs, and move issues across epics or sprints using natural language commands.

A+ Quality Score 98.33/100

agile | scrum | kanban | sprint-planning | issue-tracking

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

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.

Jira Software Cloud MCP

31 tools available
Cloud-hosted on Vinkius

Need to keep up with complex project progress? This MCP gives your AI client deep access to everything happening in Jira Software Cloud. Forget switching between dashboards and tabs; you manage the entire agile lifecycle—from initial ideas to final deployment—all through conversation. Your agent can automatically list boards, fetch all associated sprints, or find every issue linked to a specific epic. Need to adjust priorities? You can query backlogs, update issue estimations, or even submit build status and feature flag data. This integration makes Jira's complex backend logic available directly where you work. All this power is managed through Vinkius, giving your AI client one central place to access thousands of industry tools.

Core Capabilities

01 — Manage project boards

List and fetch configurations for all Scrum and Kanban boards in your account.

02 — Control sprints and timelines

Create new sprints, retrieve details, or update a sprint's status to start or close it.

03 — Audit issue backlogs and epics

Query the backlog for specific boards, retrieve all associated epics, and track issues within those large feature sets.

04 — Track development progress (DevOps)

Get build data, submit deployment statuses, or record feature flags to see real-time system health.

05 — Adjust issue priorities and status

Move issues between backlogs, epics, or sprints, and set story point estimations for accurate velocity tracking.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and provide your Jira Domain, Email, and API Token.

02 Next, connect the credentials to your preferred AI client like Cursor or Claude. Your agent now has permission to read and write data in your workspace.

03 Finally, you instruct your agent using natural language—for example, 'Start a new sprint for the Mobile App board'—and it executes the necessary Jira operations.

The bottom line is that your AI client acts as an automated layer on top of your existing Jira workflows, letting you execute complex tasks with simple text prompts.

Built For

This MCP is for technical leads and product owners who are tired of context switching. If you spend time navigating between status fields, copying issue IDs into spreadsheets, or manually checking sprint progress across multiple dashboards, this connector saves your day.

Scrum Master

You check the current state of sprints and move issues out of the backlog using simple prompts instead of clicking through the entire Jira UI.

Product Manager (PM)

You audit which epics are dragging down the timeline, rank them for priority, or retrieve issue backlogs to plan the next quarter's roadmap.

Release Engineer

You track deployments and builds associated with specific issues, submitting feature flag data directly from your agent environment.

What Changes When You Connect

01 Stop manually checking progress. Your agent can call `get_board_sprints` to instantly list all active timelines, giving you a real-time status overview without opening the board UI.

02 Keep your team moving by using `create_sprint` and `update_sprint`. You can initiate or close sprints with a single command, ensuring sprint boundaries are always accurate.

03 Improve planning depth. Instead of guessing priorities, you can use `get_board_backlog` to pull the full list of pending work, then use `rank_issues` to instantly adjust the order for the dev team.

04 Streamline DevOps tracking. Your agent handles critical release data: submitting builds via `submit_builds`, logging deployments with `submit_deployments`, and even reporting feature flag status using `submit_feature_flags`.

05 Deepen your visibility into features. By calling `get_board_epics` and then `get_epic_issues`, you get a complete audit trail of every piece of work contributing to a major product goal.

Real-World Applications

The PM needs to pivot the roadmap quickly.

A Product Manager realizes Epic 'Payment Gateway' is stalled. They ask their agent, and it uses `get_board_epics` to identify the bottleneck component. It then calls `rank_epics`, moving that epic down and prioritizing a lower-risk area using `move_issues_to_epic`. The PM gets an immediate status report and a revised plan.

The Scrum Master needs to close out the quarter.

It's time for sprint review. The agent uses `get_board_sprints` to identify all completed sprints, then calls `update_sprint` on them all simultaneously. It also runs `get_build` and `submit_incidents` to log final data before declaring the quarter finished.

The Developer needs to verify release readiness.

Before merging, a developer asks their agent about deployment status. The agent checks `get_deployment_gating_status`, then retrieves development info with `get_repository_dev_info`. If everything is green, the process continues.

The QA team needs to audit outstanding work.

QA wants to know what issues are floating in limbo. They ask their agent, which uses `get_board_backlog` to pull all pending tickets and then calls `get_issue_estimation` on the top five items so they can report readiness.

Patterns to Avoid

Trying to manage status changes manually

✗ AVOID

A user tries to update a sprint by going into Jira, changing the board view, and then clicking 'Complete'—a process that takes multiple clicks across different views.

✓ INSTEAD

Instead, ask your agent to run `update_sprint` with instructions like 'Set Sprint XYZ status to closed.' This executes the change directly through the MCP without needing UI navigation.

Overlooking dependencies during planning

✗ AVOID

A PM assigns issues to a sprint, but forgets that one of those issues belongs under a different, unmapped epic, causing scope creep.

✓ INSTEAD

Use `get_board_epics` first to review the current project structure. Then, use `move_issues_to_epic` before assigning them to ensure they are correctly grouped.

Ignoring development history

✗ AVOID

A release manager reports a feature is done, but doesn't record that the necessary builds or flags were submitted in Jira.

✓ INSTEAD

Always use `submit_builds` and `submit_feature_flags` after completing major work items. This ensures your records are complete for compliance checks.

The Right Fit

Use this MCP when your workflow requires querying, manipulating, or tracking complex, interconnected data across the entire Jira lifecycle—from initial backlog grooming to final deployment status. You need an agent that can read a board list (`list_boards`), and then use that output to trigger multiple subsequent actions (like calling `get_board_epics` followed by `move_issues_to_sprint`). Don't use this if you simply want to view one issue; just open Jira. If your goal is complex coordination, like 'Find all issues linked to Epic X that need deployment and move them to Sprint Y,' then this MCP is necessary because it orchestrates multiple tools in sequence.

The Status Update Nightmare

Today, keeping track of project status means jumping between boards. You check the backlog for new tickets; you navigate to the sprints tab to see what's due next week; then you jump over to the epic view just to confirm that a core feature is still linked to its main goal. It’s clicking through five different screens just to answer one question: 'Where are we?'

With this MCP, you ask your agent directly, 'What is the status of Epic X?' The agent runs the necessary queries—pulling data from `get_board_epics` and related tools—and spits out a single, consolidated answer. You get answers, not clicks.

Control Your Workflows with Jira Software Cloud MCP

The manual steps that vanish include: going to the board config page; finding the sprint dropdown; manually changing status from 'In Progress' to 'Ready for Review'; and then updating the epic link. All of this administrative friction is eliminated.

Your agent handles it all in one go. It lets you manage the whole flow, like calling `move_issues_to_epic` followed by `set_issue_estimation`, making your entire team's output visible through a single chat session.

Jira Software Cloud: 31 Tools

Use these tools to programmatically interact with every part of your Jira workspace, from creating boards to submitting deployment data.

#	TOOL	DESCRIPTION
01	`create_board`	This tool creates a new Scrum or Kanban board for the project.
02	`get_epic_issues`	Lists every single issue that belongs under a given epic container.
03	`submit_vulnerabilities`	Records details about security vulnerabilities found during testing or review.
04	`get_board_configuration`	Retrieves settings and structure details for any existing project board.
05	`move_issues_to_sprint`	Moves selected issues into an active or planned sprint timeline.
06	`create_sprint`	It sets up and launches an entirely new sprint timeline within Jira.
07	`get_agile_issue`	Retrieves detailed information about a specific issue, including all agile fields like story points.
08	`get_board_backlog`	Fetches the list of issues currently waiting in a board's backlog.
09	`get_board_epics`	Shows all the large, overarching feature groups (epics) attached to a specific board.
10	`get_board_sprints`	Lists every sprint that has ever been associated with this project board.
11	`get_build`	Retrieves current build status and details for the software repository.
12	`get_deployment_gating_status`	Checks if a feature is ready to deploy by checking defined gate requirements.
13	`get_epic`	Retrieves comprehensive details for a specific, high-level project epic.
14	`get_issue_estimation`	Checks the current story point estimation assigned to any individual issue.
15	`get_repository_dev_info`	Gathers general development information from the associated code repository.

#	TOOL	DESCRIPTION
16	get_sprint_issues	Shows all issues that are currently assigned to a specific, active sprint.
17	get_sprint	Retrieves the full details and status of any given sprint timeline.
18	link_security_workspaces	Connects or links necessary security workspaces required for project access.
19	list_boards	Gets a list of every board available across the entire Jira Software instance.
20	move_issues_to_backlog	Moves one or more selected issues back into the general backlog queue.
21	move_issues_to_epic	Assigns a group of issues to a specific parent epic container.
22	rank_epics	Adjusts the relative priority ranking of multiple epics against each other.
23	rank_issues	Sets or changes the priority order for a list of individual issues.
24	set_issue_estimation	Assigns or updates the story point estimate for an issue.
25	store_dev_info	Saves development information, like version numbers, to Jira for tracking.
26	submit_builds	Reports and submits build data, indicating when a new code compilation is ready.
27	submit_deployments	Logs the successful completion of a software deployment to an environment.
28	submit_feature_flags	Reports the status of feature flags, indicating if code features are live or hidden.
29	submit_incidents	Records details about system incidents or formal reviews that occurred during development.
30	submit_remote_links	Adds links to external documentation or resources relevant to the issue.
31	update_sprint	Changes the status of a sprint, marking it as active or completed.

See It in Action

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

U List all available Jira boards.

I've found 3 boards: 'Mobile App' (ID: 1), 'Web Backend' (ID: 2), and 'Marketing' (ID: 3). Which one would you like to explore?

U Show me the issues in the backlog for board 1.

Retrieving backlog for 'Mobile App'... I found 12 issues. The top items are 'Fix login bug' and 'Update splash screen'. Would you like more details on any of these?

U What is the estimation for issue 10123?

The estimation for issue 10123 is currently set to 5 story points.

Frequently Asked Questions

01 **How do I check which issues are part of an epic using Jira Software Cloud MCP?**

You use the `get_epic_issues` tool. This function takes an Epic ID and returns a comprehensive list of every issue that belongs under it, letting you audit the scope immediately.

02 **Can I automatically start or end a sprint with Jira Software Cloud MCP?**

Yes, use the `update_sprint` tool. You can pass instructions to change the sprint's state—marking it active or closed—without needing access to the UI.

03 **Does this MCP help with DevOps tracking for my project?**

Absolutely. The MCP provides tools like `get_build` and `submit_deployments`. You can track build status, submit deployment logs, and check feature flag settings from your agent.

04 **What if I need to change the priority of a bunch of issues?**

You use the `rank_issues` tool. Simply tell your agent which issues you want to rank and in what order, and it updates them across the board.

05 **Can Jira Software Cloud MCP help me organize tickets into a new epic?**

Yes. You use `move_issues_to_epic`. This tool allows you to select multiple issues and assign them as a group to a specific parent epic.

06 **How do I view the backlog for a specific board?**

Use the `get_board_backlog` tool with the target `boardId`. It will return all issues currently in the backlog for that Scrum or Kanban board.

07 **Can I change a sprint's status to active or closed?**

Yes, use the `update_sprint` tool and provide the `sprintId` along with the desired `state` ('active' or 'closed').

08 **Is it possible to check the story point estimation for an issue?**

Yes! The `get_issue_estimation` tool allows you to retrieve the estimation value for any specific agile issue using its ID.

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 `"jira-software-cloud": { "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

Jira Software Cloud 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 Jira Software Cloud. 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	Jira Software Cloud MCP
Server ID	`019e38b0-53a7-7228-9b5b-2f4c9222cb65`
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/jira-software-cloud.