MCP SERVER | NO CODE | CLOUD HOSTED

DVC MCP for AI Agents

Manage ML experiment tracking and data versioning history

DVC MCP connects your AI agent directly to your DVC Studio account for ML experiments. Stop clicking through dashboards and start asking natural language questions about model runs, project history, and data metrics. Audit projects, track views, and manage the entire lifecycle of your machine learning models via conversation.

A+ Quality Score 100/100

machine-learning | version-control | experiment-tracking | data-pipelines | model-management | git-workflow

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

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.

DVC MCP

6 tools available
Cloud-hosted on Vinkius

Managing large-scale ML projects usually means jumping between a dozen different tabs: the dashboard, the Git repo, the metric logging service. It's slow, tedious, and prone to human error.

This MCP changes that. You connect your DVC Studio credentials once, and your AI client gets direct access to your entire data versioning workflow. Instead of manually navigating complex project structures or searching through log files for a specific accuracy score, you just ask your agent what you need.

You can tell it to list all active projects, check the history of model runs, or pull up structural details about dashboard views—all in plain English. It’s like having an expert ML Ops engineer sitting next to you, ready to answer any question about project data and versioning without ever leaving your chat window. This capability is available through Vinkius, making it easy to connect this core function into whatever AI client you already use.

Core Capabilities

01 — List all dashboard views

Retrieves a list of defined UI configuration layouts within your DVC Studio workspace.

02 — Retrieve specific view details

Fetches the structural settings and configurations for a single, chosen dashboard view.

03 — Get user profile information

Retrieves basic metadata about the authorized user account connected to DVC Studio.

04 — List all active projects

Provides a list of registered organization workspaces available within your DVC Studio environment.

05 — Get specific project details

Retrieves the full metadata and current status for an individual, specified ML project.

06 — List all model experiments

Generates a list of completed or running machine learning experiment runs tied to a specific project.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your DVC Studio Client Access Token. This token grants the AI client permission to read your ML project data.

02 Tell your agent what you need, for example: 'Show me all projects I've set up.' The agent translates that request into a structured query for your DVC account.

03 The system executes the query and returns the specific, requested information—like a list of model runs or project metadata—directly back to your chat interface.

The bottom line is, you talk naturally about complex ML data versioning tasks, and this MCP handles the technical calls behind the scenes.

Built For

This MCP is for experienced Data Scientists and MLOps Engineers who are tired of spending hours clicking through dashboards and manually cross-referencing logs. If your job involves tracking hundreds of model versions or auditing experiment failures, this tool saves you serious time.

Machine Learning Engineer

Uses the MCP to audit model run histories by asking for specific metrics arrays from past experiments, verifying that dependencies are correctly versioned.

Data Scientist

Uses this tool daily to monitor active projects and list all available views when starting a new experiment, ensuring the right dashboard layout is used immediately.

Team Lead / MLOps Manager

Checks organization workspaces and lists projects across different team members in natural language conversation, keeping everyone aligned on progress without digging into dashboards.

What Changes When You Connect

01 Instead of navigating through the DVC Studio UI to find model runs, you simply ask your agent to list experiments. This gives you instant access to run IDs, completion statuses, and performance summaries.

02 You don't have to guess which dashboard layout is correct. Use the `list_views` tool to see all active views, then use `get_view` to pull up detailed structural settings for a specific one.

03 Audit your entire ML portfolio easily. You can list projects and get full metadata on any workspace using `get_project`, helping you quickly verify if a dependency exists before starting work.

04 Deep dive into performance metrics without manual logging searches. Your agent lets you request complex structural arrays defining exactly which metrics were captured during specific experiment epochs.

05 Keep track of who's doing what. Get the current user profile using `get_user` to verify permissions and identify the authorized token holder when collaborating with a team member.

06 When troubleshooting, quickly see all possible model experiments by calling `list_experiments`, giving you an immediate overview of the entire project history.

Real-World Applications

Finding the Best Model Run from Last Quarter

A data scientist needs to know which specific run achieved the highest accuracy for a fraud detection model. Instead of opening the dashboard and filtering by date, they ask their agent to list experiments, providing IDs and peak metrics immediately.

Verifying Project Scope Before Development

An ML engineer starts a new task but needs confirmation that all required data sources are accounted for. They ask the agent to list projects across their organization to verify repository connections against internal team mappings, preventing build failures.

Understanding Dashboard Configuration Changes

A team lead takes over a project and needs to know what metrics were tracked previously. Instead of clicking through the dashboard settings repeatedly, they ask their agent to list views and retrieve detailed configurations for all existing dashboards.

Auditing Compliance and Access Rights

A DevOps engineer must confirm that only authorized personnel have access to sensitive model data. They use the agent to get user profile information and audit project metadata to verify current permissions against security guidelines.

Patterns to Avoid

Assuming all projects are visible

✗ AVOID

A developer tries to troubleshoot an issue in a new workspace but doesn't know the exact name, leading them to manually check every single folder and repository link.

✓ INSTEAD

First, use the `list_projects` tool to get a comprehensive list of all workspaces. Then, call `get_project` with the correct identifier to pull up the full metadata and verify connections.

Getting lost in metric logs

✗ AVOID

A data scientist needs to know if a specific metric (like AUC) was tracked during an experiment but spends hours scrolling through unstructured log files.

✓ INSTEAD

Use the agent to request complex structural arrays defining metrics. This tells you precisely which performance indicators were captured for any given model run.

Forgetting existing dashboard layouts

✗ AVOID

A team member needs to update a key metric display but can't find the original dashboard configuration, resulting in hours of recreating widgets and settings.

✓ INSTEAD

Always start by calling `list_views`. This instantly shows all existing UI configurations. You can then use `get_view` to pull up the detailed setup for the specific layout you want to edit.

The Right Fit

Use this MCP if your ML workflow relies on constant auditing, cross-referencing project metadata, or tracking complex model version histories. It's essential when your job requires converting unstructured questions about 'Why did Model X perform poorly?' into concrete data retrieval calls (like listing experiments or getting view details).

Don't use this if you simply need to manage basic file transfers or simple database queries; a general-purpose data connector will work better. Also, don't expect it to train models—it only tracks and reports on them. If your goal is just to check user access tokens, the `get_user` tool provides that specific metadata, making it excellent for compliance checks.

DVC MCP: Tracking ML Experiment History with Conversational AI

Right now, tracking model performance is a painful clicking ritual. You have to open the DVC Studio UI, navigate into the project workspace, manually select 'History,' and then filter by date range or run ID just to see which metrics were captured for comparison. It's slow, it requires too many context switches, and you often miss key details buried in the configuration settings.

With this MCP, the process flips entirely. You simply ask your agent: 'Show me all projects that have completed runs with an accuracy above 0.9.' The agent handles the complex navigation, compiles the list of experiments, and presents the results immediately, giving you a single pane of glass view of your entire model portfolio.

DVC MCP: Managing Project Dependencies via Natural Language

Before starting any major iteration, most engineers spend time verifying the project's full scope. This means going through multiple tabs to list active projects, checking repository connectivity for each one, and confirming that all necessary dashboard views are correctly set up—a process ripe for human error.

Now you just ask your agent: 'What are my current organization workspaces and what dashboards do they use?' It runs `list_projects` and `list_views` in sequence. You get a structured, immediate answer detailing the entire project scope without opening a single browser tab.

6 Tools for ML Experiment & Project Tracking

Use these tools to list projects, views, and retrieve specific metadata about model runs and workspace configurations.

#	TOOL	DESCRIPTION
01	`list_views`	Lists all defined dashboard views currently available in your DVC Studio account.
02	`get_view`	Retrieves the detailed configuration and structural settings for a specific dashboard view.
03	`get_user`	Returns basic profile information about the connected DVC Studio user account.
04	`list_projects`	Retrieves a list of all registered ML projects (organizational workspaces) managed by your DVC Studio account.
05	`get_project`	Fetches detailed metadata and status for one specific project identifier.
06	`list_experiments`	Lists all recorded model experiments, showing key identifiers and run statuses within a given project.

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 top 3 projects I ran last month?

Project Summary: Last 3 Months

PROJECT NAME	BEST METRIC (ACCURACY)	RUN ID
Credit-Scoring-Model	0.942	exp-a1b2
Image-Classification-V2	0.875	exp-c3d4
NLP-Pipeline	0.910	exp-x9yz

Need to see the specific metrics array for any of these? Just ask.

U Can you list all dashboard views and what they contain?

You have 2 active dashboard layouts:

1. **Main-Metrics-Grid**: Displays primary KPIs (Accuracy, F1 Score) across a grid format.
2. **Validation-Plots**: Shows detailed plot structures for ROC curves and confusion matrices.

You can get the full settings for either one if you want to modify it.

U What are my current user permissions in DVC?

DVC Studio User Profile

- **User:** Jane Doe (janedoe@corp.com)
- **Role:** Data Scientist
- **Scopes:** Read/Write access for all projects under the 'Finance' organization scope.
- **Token Validity:** Active, expires 2026-12-31.

Frequently Asked Questions

01 **How does DVC MCP help me track my model experiments?**

It lets you use natural language to audit your entire experiment history. Instead of clicking through dashboards, you can ask for specific metrics arrays or list all runs just by talking to your AI client.

02 **Can I find out what projects my team has set up?**

Yes. You simply ask the MCP to list all active projects. It gives you a clear overview of every organizational workspace, helping you manage dependencies and understand the scope of work.

03 **Is this better than just using the DVC Studio web interface?**

It's faster because it eliminates clicks. Instead of navigating multiple menus to find a specific project or view, your agent retrieves that data directly into the chat window in seconds.

04 **What kind of information can I get about dashboard views?**

You can list all available views and retrieve their structural settings. This is great for checking if a metric was tracked correctly or verifying which widgets are active on any given board.

05 **How do I verify my permissions using DVC MCP?**

If you need to check who has access or what scopes your token covers, you ask the agent for user profile information. This gives you a quick audit of authorized roles and tokens.

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 `"dvc": { "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

DVC 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 DVC. 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	DVC MCP
Server ID	`019d758a-bbf3-7278-9ad6-5ea027c24660`
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/dvc.