MCP SERVER | NO CODE | CLOUD HOSTED

Anyscale MCP for AI Agents

Manage MLOps Cluster Jobs and Model Inference

The Anyscale MCP lets your AI client manage entire distributed machine learning environments through natural conversation. You can list models, generate vector embeddings for large text arrays, monitor deployed services, and check complex Ray cluster job statuses—all without opening a terminal or navigating a heavy cloud dashboard.

F Quality Score 43.65/100

distributed-computing | llm-inference | vector-embeddings | cluster-management | scalable-ai

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

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.

Anyscale MCP

7 tools available
Cloud-hosted on Vinkius

This connector connects your AI agent directly to the Anyscale environment, letting you manage both large-scale LLM queries and underlying backend infrastructure natively. Instead of logging into a clunky web portal just to check if a training job finished, you talk to your agent. It handles the complex background work for you.

It provides tools to list active foundational models and run chat completions using specialized Anyscale LLMs. You can also generate semantic vector embeddings from text inputs on the fly. Furthermore, it lets you monitor deployed Ray services and query batch jobs to inspect their recent execution statuses and training metrics via conversation. If you're already using Vinkius for your other APIs, adding this MCP gives you a single point of control over your entire MLOps stack.

Core Capabilities

01 — list_models

Lists all foundational AI models available on your Anyscale Endpoints cluster.

02 — chat_completion

Generates conversational replies by sending structured messages with roles (user, system, assistant) to Anyscale LLMs.

03 — text_completion

Creates text completions using the general Anyscale API when you need foundational, non-conversational text generation.

04 — generate_embeddings

Processes arrays of text and generates semantic vector embeddings that can be used for advanced search or RAG systems.

05 — list_services

Retrieves an overview list of all currently deployed services on your Anyscale platform.

06 — get_service

Fetches specific, detailed information about a single designated Anyscale service deployment.

07 —

Lists all running or completed batch and training jobs managed by your Ray cluster on Anyscale.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP, providing your specific Anyscale API Key and Base URL.

02 Connect your preferred AI client (like Cursor or Claude) to the Vinkius catalog using your credentials.

03 Ask your agent to perform tasks—for example, 'What's the status of my latest training job?' The agent then invokes the necessary tools.

The bottom line is, you get a conversational layer over highly technical ML infrastructure management.

Built For

This MCP targets MLOps Engineers and Data Scientists who struggle with context switching. If your job involves monitoring deployed models or running large-scale batch processing without constantly opening clunky terminal dashboards, this is for you.

MLOps Engineer

You use it to safely automate the inspection of model deployment status and background jobs during CI/CD workflows.

Data Scientist

You submit rapid, specialized completion tasks to LLMs running inside your private Anyscale VPC for research or prototyping.

Backend Developer

You debug service health metrics and endpoint statuses quickly without having to navigate the heavy cloud dashboard UI.

What Changes When You Connect

01 You can check the status of large-scale training jobs using the `list_jobs` tool, getting execution metrics without opening a separate terminal window.

02 Instead of manually checking multiple dashboards, you use the MCP to list all active models (`list_models`) and confirm they are ready for inference immediately.

03 Generating vectors is fast. The `generate_embeddings` capability processes large text arrays directly, which is critical for building RAG pipelines efficiently.

04 Debugging service issues is simpler. You just need to use the MCP's `get_service` function to pull up specific endpoint details in a conversation.

05 The ability to run conversational queries (`chat_completion`) means you interact with complex model outputs using plain language prompts, not API JSON structures.

Real-World Applications

Checking Model Readiness After Deployment

An MLOps Engineer needs to validate that a newly trained LLM is live. Instead of logging into the console dashboard and waiting for status lights to turn green, they ask their agent to list models, confirming the exact model ID is available for use.

Retrieving Training Metrics Mid-Run

A Data Scientist notices a job slowing down. Instead of searching through historical logs, they tell their agent to query the latest jobs, immediately seeing if the 'daily_retrain' run completed successfully or failed on specific nodes.

Building a Search Index for Documentation

A developer needs to index hundreds of technical documents. They use the MCP to generate vector embeddings for all text, feeding them directly into their data pipeline rather than running a separate embedding service script.

Validating Service Health Before Go-Live

A Backend Developer needs to ensure a specific microservice is healthy before traffic hits it. They use the agent's ability to retrieve details about a specific service, confirming endpoint configurations and operational status.

Patterns to Avoid

Treating LLMs like general search engines

✗ AVOID

Asking your AI client vague questions that require it to guess the underlying cluster state or model availability.

✓ INSTEAD

Be specific. Use the MCP to first list models with `list_models` and then ask for chat completions using a known, verified model name.

Ignoring job status checks

✗ AVOID

Assuming that because you submitted a training batch job, it's automatically finished and ready for the next step.

✓ INSTEAD

Always use `list_jobs` to query the current execution statuses. This confirms if the job succeeded or failed before moving forward.

Overcomplicating vector creation

✗ AVOID

Trying to manually split large documents into chunks and then running embedding generation for each chunk sequentially.

✓ INSTEAD

Use `generate_embeddings` on the full text array. The MCP handles the efficient processing of large batches, saving time.

The Right Fit

Use this Anyscale MCP if your core pain point is coordinating complex MLOps tasks across multiple tools and dashboards. You need a single conversational interface to check job status (`list_jobs`), validate model availability (`list_models`), and handle foundational data prep like vector generation. Don't use it if you only need simple text completion; for that, a standalone LLM API might suffice. If your workflow is entirely self-contained (e.g., just running a single, isolated script), this MCP adds unnecessary overhead. But if you manage distributed compute, service endpoints, and multiple AI models, this tool saves significant time.

Anyscale MCP for AI Agents: Managing MLOps Cluster Jobs

Today, checking the status of a batch job or validating model deployment involves jumping between three different places: the Ray cluster dashboard, the service registry UI, and the logs viewer. You copy statuses from one place into a spreadsheet just to track failures.

With this MCP, you simply tell your agent what you need to know about the cluster jobs. It queries the necessary services behind the scenes, pulling the execution status and training metrics directly into the chat interface. The result is an immediate answer, not a link to three different dashboards.

Anyscale MCP for AI Agents: Controlling Model Inference Workflows

Running complex LLM queries means juggling model names, API keys, and whether the required models (like Llama-2) are actually deployed. You spend time verifying if the foundational model is available before you can even start writing your prompt.

This MCP gives you instant visibility via `list_models`. It shows every single active model ready to receive inference traffic, confirming deployment status in one quick conversation. That immediate confirmation keeps your workflow moving without delay.

Anyscale MCP: 7 Tools for Vector Embeddings & Cluster Management

These tools let you manage everything from listing foundational AI models to running complex batch jobs and generating vector embeddings, all within a conversational flow.

#	TOOL	DESCRIPTION
01	`list_models`	Lists all foundational AI models available on your Anyscale Endpoints cluster.
02	`chat_completion`	Generates conversational replies by sending structured messages with roles (user, system, assistant) to Anyscale LLMs.
03	`text_completion`	Creates text completions using the general Anyscale API when you need foundational, non-conversational text generation.
04	`generate_embeddings`	Takes a piece of text and creates its corresponding semantic vector embedding array.
05	`list_services`	Retrieves an overview list of all currently deployed services on your Anyscale platform.
06	`get_service`	Fetches specific, detailed information about a single designated Anyscale service deployment.
07	`list_jobs`	Lists all running or completed batch and training jobs managed by your Ray cluster on Anyscale.

See It in Action

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

U List all the foundational models I have deployed right now.

✅ **Models Found:**

- `meta-llama/Llama-2-70b-chat-hf` : Active. Type: Chat.
- `mistralai/Mistral-7B-Instruct-v0.1` : Active. Type: Chat.
- `openai/gpt-4o` : Deprecated. Status: Pending Update.

U What's the status of my last two training jobs?

I checked your recent cluster activity. Here are the details:

JOB NAME	STATUS	COMPLETION TIME
daily_retrain_v3	✅ Succeeded	5 hours ago
experimental_tuning_7b	❌ Failed	Out of memory (Node 12)

You need to check the service metadata for Node 12.

U Can you generate an embedding for this paragraph?

Sure thing. I passed the text to your configured model. The resulting array has 768 dimensions and is ready to use in your data pipeline. Here's a snippet of the first few floats: `[0.12, -0.55, 0.91, ..., 0.34]`. Let me know if you need it mapped somewhere specific.

Frequently Asked Questions

01 **How does the Anyscale MCP help me check my cluster job status?**

The Anyscale MCP lets you query your Ray batch jobs directly through conversation. Instead of opening a complex terminal dashboard, simply ask about recent job statuses to see if training succeeded or failed and why.

02 **I need to find out which LLMs are available on my cluster using the Anyscale MCP?**

You can use the MCP to list all active foundational models. It gives you a clean rundown of every deployed model, confirming its name and current status before you write a single line of code.

03 **What if my service endpoint is having issues? Can Anyscale MCP help me debug it?**

Yes, the MCP allows you to retrieve specific details about your deployed services. This means you can confirm the latest endpoint configurations and check the current health status of a microservice in plain language.

04 **Does Anyscale MCP handle generating embeddings for my documents?**

It does. You pass text to the MCP, and it generates semantic vector embeddings using your configured model. This makes preparing data for search or RAG pipelines much easier than running separate scripts.

05 **How do I connect Anyscale MCP to my AI agent?**

You subscribe to this MCP in the Vinkius catalog, providing your necessary Anyscale API keys. Your agent then handles all the communication with the cluster tools for you.

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 `"anyscale": { "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

Anyscale 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 Anyscale. 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	Anyscale MCP
Server ID	`019d754e-a2ee-73d3-8d87-cd2019c58c1a`
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/anyscale.