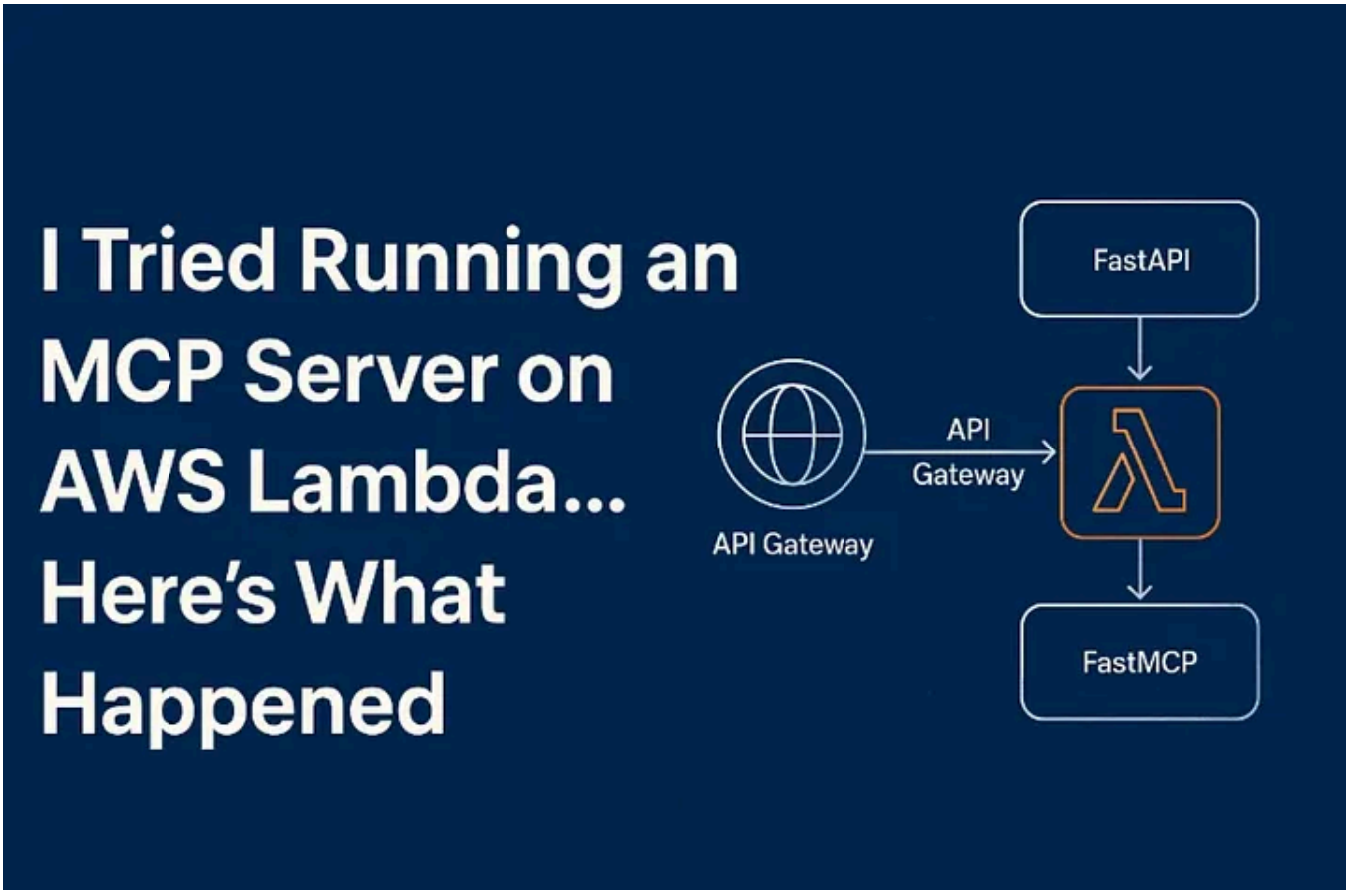
Member-only story

I Tried Running an MCP Server on AWS Lambda... Here's What Happened

5 min read · May 10, 2025

Ran Isenberg Follow

Listen Share More



I Tried Running an MCP Server on AWS Lambda... Here's What Happened

Two days ago, the official MCP Python SDK released support for **streamable HTTP transport** via **FastMCP**, which also supports FastAPI. As someone deeply involved in

building secure and production-grade serverless services, I wanted to test the viability of running an MCP server *serverlessly* on AWS Lambda.

Let's just say — **we're not there yet**.

In this brief blog post, I'll share my experience and thoughts on the current state of serverless MCP servers.

July 16th update — Check out the new & updated blog post where I compare THREE ways to deploy MCP on AWS including GitHub template repo — https://www.ranthebuilder.cloud/post/serverless-mcp-on-aws-lambda-vs-fargate-for-agentic-ai-workloads

UPDATE: AS of June 11th 2025 — I have created The AWS Lambda MCP Cookbook template — a pure Lambda, no FastMCP or other hacks implementation with:

- **Python Serverless MCP server with a recommended file structure.**
- **Tests — unit, integration (tests for full MCP messages) and E2E with a real MCP client**
- **CDK infrastructure with infrastructure tests and security tests.**
- **CI/CD pipelines based on Github actions that deploys to AWS with python linters, complexity checks and style formatters.**
- **CI/CD pipeline deploys to dev/staging and production environments with different gates between each environment**
- **Makefile for simple developer experience.**
- **The AWS Lambda handler embodies Serverless best practices and has all the bells and whistles for a proper production ready handler.**
- **AWS Lambda handler uses AWS Lambda Powertools.**
- **AWS Lambda handler 3 layer architecture: handler layer, logic layer and data access layer**
- **Session context storage in DynamoDB (does NOT send it to tools yet)**
- **API protected by WAF with four AWS managed rules in production deployment**

- **CloudWatch dashboards — High level and low level including CloudWatch alarms**

Check out the GitHub repo!

https://www.ranthebuilder.cloud/

Book AWS Serverless or Platform Engineering consultation today!

. . .

Table of Contents

. . .

Why MCP on Lambda?

The real question should be, **why not**?

MCP (Model Context Protocol) is an open standard that aims to simplify how AI agents communicate with tools and resources. With its new HTTP transport support, it theoretically becomes deployable anywhere HTTP servers can run, like FastAPI.

But as you know, I'm an AWS serverless hero, so naturally, combining MCP, FastAPI, and AWS Lambda (with the Lambda Web Adapter) felt like an interesting experiment.

My architecture looked like this:

```
API Gateway → Lambda Web Adapter → FastAPI → FastMCP
```

FastAPI handles routing and server logic. FastMCP exposes the resources. The AWS Lambda Web Adapter (repo) enables this by allowing HTTP servers to run inside Lambda using a custom container image. I used my blog post about building Lambda functions from container images (which you can read here), and we were off to the races.

Dockerfile:

https://gist.github.com/ran-isenberg/154074c6bff01e9cbedcd53590b1c2c1

If you're curious to try it yourself, AWS published code samples: Stateless MCP on Lambda (Python). It uses SAM. I used CDK.

. . .

Developer Experience: Rough and Raw

Let's talk DevEx. It's confusing, inconsistent, and far from intuitive — but that's somewhat expected for a just-released integration. Still, here are the pain points:

IaC Support

It's not trivial at all. You need to figure out many things yourself and configure all the moving parts together, based on some documentation hints. ChatGPT and the like were not helpful at all.

Performance

Cold starts are brutal. I experienced delays of **up to 5 seconds and had to bump up the memory**. That's unsurprising given the number of moving parts (you're spinning up a whole server!), But this setup is still unsuitable for latency-sensitive workloads.

Observability

Logging formats are all over the place. FastAPI and FastMCP each have their own loggers and settings. I use Powertools for AWS Lambda for logging.

I tried redirecting all logs through Power Tools for structured JSON logging, but failed to get it working (I haven't dug in enough yet).

In addition, you don't get access to the usual event and context objects since you're not writing a Lambda handler but spinning up a server. That breaks **Powertools utilities** like tracer and metrics. You can have tracing, but you can have inner handler/function segments at least, not with Powertools native decorators. I'm sure there's a way to get it working with some middleware for FastAPI.

some are JSON, most are not

No Standard Lambda Experience

This setup throws away the clean simplicity of writing a Lambda handler. You're running a server now, which means no easy integration with your typical Lambda-native tools like PowerTool's event handler. If you're used to event-driven Lambda development, this feels like stepping back in time. It just doesn't fit the handler-logic-DAL architectural layers formats (hexagonal arch.) we've been using with Lambda.

Testing

I had to manually configure VS Code to interact with the MCP server and send the correct payload — something I couldn't figure out without digging into the protocol's RFC.

You can run the server in stdio mode for local development, which helps. However, for testing, I wanted to construct requests directly in **Pytest** rather than relying on an MCP client (as most examples do). The current experience lacks developer-friendly wrappers and tooling.

It's doable — but far from seamless.

. . .

Comparing to Bedrock Agents on Lambda

Last July (this was 2024, mind you!), I wrote a blog post about using Bedrock agents with OpenAPI on Lambda. Within **30 minutes**, I had a production-grade setup using:

- Lambda handler with API GW + WAF
- AWS Powertools (logging, tracing, metrics)
- Excellent performance
- Chat with an agent that triggered my APIs

So what's the difference?

Bedrock used proprietary protocol AWS built on-top of HTTP.

MCP is an open standard protocol with additional abstraction layers, BUT it's universally hyped and accepted.

It feels like a classic case of **"one step forward, two steps back."** But it's a step nonetheless, and I'm hopeful that tooling and performance will improve.

. . .

What Needs to Improve

Everything.

To the **AWS Lambda and Powertools for AWS Lambda teams**, if you're reading this:

- We need **first-class observability support** (logging, tracing, metrics)
- Cold start performance needs drastic optimization
- Easier setup and full IaC support.

Until then, this is a cool toy experiment, not production-ready.

. . .

The Future

I took the liberty to open a PR with the ideal DevEx I had in mind for the Powertools team:

https://github.com/aws-powertools/powertools-lambda-python/issues/6562

This is simple. We use define a handler and the MCP tools and it just **WORKS.**

I hope this post will age very quickly and soon!

AI | Mcp Server | Serverless | AWS | AWS Lambda

Follow

Written by Ran Isenberg

1.5K followers · 39 following

AWS Serverless Hero | Principal Software Architect @CyberArk | Consultant ranthebuilder.cloud

Responses (5)

Bgerby

What are your thoughts?

Patrizio Bruno
May 10 (edited)

Just reading your comment about one of the biggest problems being running a server inside a serverless function made me think about Azure Container Apps. It seems a better fit for running a full fledged stateful web application instead of ephemeral functions.

7 2 replies Reply

Bob Seaton
Jun 6

> can

should this be "can't"?

3 Reply

William Claude25 him
Sep 17

Let’s just say — we’re not there yet.

1 reply Reply

See all responses

More from Ran Isenberg

In AWS in Plain English by Ran Isenberg

Serverless MCP on AWS: Lambda vs. Fargate for Agentic AI Workloads

Learn how to build MCP servers on AWS for agentic AI. Compare Lambda, Web Adapter, and Fargate options with CDK templates and real-world...

Jul 16

In AWS in Plain English by Ran Isenberg

Secrets Manager vs. Parameter Store: Which One Should You Really Use?

Compare AWS Secrets Manager vs. SSM Parameter Store: costs, rotation, versioning, IaC, and use cases—learn when to use each based on...

Sep 8

In AWS in Plain English by Ran Isenberg

AWS re:Invent 2025—My Selection Of Sessions—Serverless, Security, SaaS and AI

In this post, you will find my opinionated list of AWS re:Invent 2025 breakout sessions, workshops, builder sessions, code-talks, dev...

Oct 8

Ran Isenberg

Amazon SQS Dead Letter Queues and Failures Handling Best Practices

Amazon Simple Queue Service (SQS) is a powerful service designed to manage the messaging needs of robust, decoupled microservices.

Aug 15, 2023

See all from Ran Isenberg

Recommended from Medium

In Data, AI and Beyond by Julius Nyerere Nyambok

Generate AWS architectural diagrams using this simple method

A combination of LLMs and MCP Servers.

Oct 14

In AWS in Plain English by Aman Pathak | DevOps | AWS | K8s | Terraform

AWS Outage Root Cause Revealed: How a DNS Mismatch Affected DynamoDB and Core Services

Discover what caused the October 2025 AWS us-east-1 outage—a DNS mismatch that broke DynamoDB and triggered cascading failures across...

Oct 21

Neal Davis

Cloud Skills that will soon be obsolete—and what to learn instead

If you're learning cloud right now—pay attention. Some of the most common skills people still spend hours practicing are already...

Oct 13

Heeki Park

Building an MCP server as an API developer

Anthropic released MCP at the end of November 2024. It took a few months for it to catch on, but my, oh my, it feels like the community is...

May 14

In Dare To Be Better by Max Petrusenko

Claude Skills: The $3 Automation Secret That's Making Enterprise Teams Look Like Wizards

How a simple folder is replacing $50K consultants and saving companies literal days of work

Oct 17

Anil Kumar Nayak

Building an AI Agent That Debugs Production Incidents

From Tab-Hopping at 2 AM to AI-Powered Root Cause in Seconds

Aug 23