

# Mehdi Salhi

## Cloud & DevOps Engineer

mehdisalhi.dev@gmail.com | (718) 500-1191 | Stamford, CT | U.S. Permanent Resident (no sponsorship needed)  
linkedin.com/in/mehdi-salhi-work | github.com/medysaly | mehdisalhi.com

### SUMMARY

Computer Science student at SNHU (graduating November 2026, GPA 3.71), AWS Certified Solutions Architect Associate and Cloud Practitioner. Experienced designing, deploying, and securing production-grade systems on AWS, from a serverless FinOps tool that surfaced real cost savings to AI applications on Amazon Bedrock (LLMs, RAG, and agent workflows). Skilled with Python, Linux, Terraform, Docker, and secure CI/CD pipelines (GitHub Actions and OIDC), and currently building with Kubernetes. Seeking a Cloud or DevOps internship or entry-level role.

### TECHNICAL SKILLS

**Cloud & Infrastructure (AWS):** Lambda, EC2, S3, DynamoDB, API Gateway, EventBridge, VPC, IAM, CloudWatch

**IaC, CI/CD & Containers:** Terraform, AWS SAM, CloudFormation, Docker, Kubernetes, GitHub Actions, OIDC

**Languages & Tools:** Python, Bash, Linux, Git, YAML, SQL

**AI on AWS:** Amazon Bedrock, Amazon SageMaker, RAG, LLMs, Prompt Engineering

### EXPERIENCE

#### AI Agent Engineering Extern - Extern (Wayfair n8n AI Agent Engineering Externship)

*Remote · July 2026 - Present*

- Building AI-agent workflows in n8n that research product trends across Amazon, Instagram, and Pinterest and surface top-selling products to a data dashboard.
- Automating multi-source data collection and trend analysis with AI agents, LLMs, and API integrations.

### PROJECTS

#### AWS Cost Watchdog - Serverless FinOps monitoring tool

*Terraform · Lambda · EventBridge Scheduler · SNS · DynamoDB · API Gateway · Python · React | github.com/medysaly/aws-cost-watchdog*

- Automated daily AWS cost and waste monitoring, spend digests, idle-resource detection, tag-compliance checks (AWS Config), and cost-anomaly detection, pushing every finding to Slack and Telegram in real time, across four event-driven Python Lambdas on EventBridge Scheduler (cron) and SNS.
- Persisted findings to DynamoDB and surfaced them in a React dashboard served through an API Gateway HTTP API and S3/CloudFront, for at-a-glance spend and waste visibility.
- Provisioned the whole stack as infrastructure as code with Terraform (remote S3 state and DynamoDB locking) and shipped it via a GitHub Actions CI/CD pipeline using OIDC federation, with zero static AWS credentials.

#### StockWatch - Serverless AI market-brief on AWS

*Python · Lambda (container/ARM64) · ECR · EventBridge · Terraform · GitHub Actions (OIDC) · Claude (Anthropic API) | github.com/medysaly/stockwatch*

- Automated a daily AI market brief that pulls price and news data (yfinance) and summarizes it with Claude, running hands-free on an EventBridge daily schedule at a few dollars per month.
- Guarded the LLM output with automated pytest checks (non-empty, no refusals) enforced in a GitHub Actions pipeline alongside ruff linting and a terraform plan gate.
- Deployed as an ARM64 container image on ECR with zero static AWS credentials (GitHub OIDC), and stored API keys in a single Secrets Manager secret behind a least-privilege IAM policy.

#### Unkonmon - Full-stack AI website on a serverless AWS backend

*React/TS · Lambda · API Gateway · DynamoDB · Bedrock (Claude Haiku 4.5) · AWS SAM · WAF | github.com/medysaly/unkonmon · unkonmon.ai*

- Built and deployed Unkonmon end to end, the website and serverless AWS backend: a React site with an AI chatbot and a VAPI voice receptionist that answer questions, book appointments, and capture leads, across three Lambdas behind API Gateway (IaC with AWS SAM).
- Reduced chatbot latency and Bedrock spend by front-running a Trie-based intent classifier ahead of Claude Haiku 4.5 streaming responses, so common questions never hit the model.
- Hardened the backend with a WAFv2 web ACL, HMAC-verified webhooks, least-privilege IAM, and Secrets Manager, over a DynamoDB data layer with Global Secondary Indexes, all tested through a GitHub Actions pytest pipeline.

#### Company Policy RAG - Retrieval-Augmented Generation over policy documents

*Python · FastAPI · Streamlit · BM25 + dense retrieval · Cross-encoder reranker · RAGAS · Docker | github.com/medysaly/company-policy-rag*

- Built a RAG pipeline over policy documents with hybrid retrieval (dense embeddings and BM25) and cross-encoder reranking, measured with RAGAS at 1.00 faithfulness and 1.00 context precision on a 10-question evaluation set.
- Engineered the ingestion and chunking pipeline with unit tests and a GitHub Actions CI pipeline, documenting the key retrieval design tradeoffs.
- Served it through a FastAPI backend with a Streamlit UI, containerized with Docker and deployed on Hugging Face Spaces.

### CERTIFICATIONS

AWS Certified Solutions Architect - Associate (2026) • AWS Certified Cloud Practitioner (2026) • Google IT Support Professional Certificate (2026)

### EDUCATION

**B.S. in Computer Science**, Southern New Hampshire University • Expected November 2026 • GPA 3.71