

Christian Santiago

Cloud/Software Engineer moving towards MLOps – AWS Infrastructure, Data Pipelines, Cost Efficiency, Service Reliability
santiagothedeveloper@gmail.com | christiansantiago.dev | linkedin.com/in/christian-santiago-dev | github.com/Go-Santiago-Go

EDUCATION

Vanderbilt University – Computer Science	Nashville, TN
AWS Cloud Institute – Cloud Application Developer	Remote
Codesmith – Software Engineering + AI/ML Immersive	Remote

EXPERIENCE

Amazon Web Services Cloud Institute

Cloud Engineering Resident

Sep. 2025 – Jul. 2026

- Developed cloud application skills through hands-on AWS labs across serverless and container services, building event-driven workflows with AWS Lambda, Amazon API Gateway, Amazon DynamoDB, Amazon S3, Amazon SNS, and Amazon SQS, orchestrating multi-step processes with AWS Step Functions, and deploying containerized applications with Docker, Amazon ECS, Amazon EKS, and Amazon ECR.
- Applied AWS AI/ML services including Amazon Bedrock, Amazon Rekognition, and Amazon Textract, and built Retrieval Augmented Generation (RAG) workflows with LangChain, practicing Infrastructure as Code (IaC) and CI/CD, with least-privilege IAM and observability through Amazon CloudWatch and AWS X-Ray.

Price Printing

Business Systems Analyst

May 2025 – Feb. 2026

- Built and maintained ETL pipelines standardizing multi-channel client-provided mailing address data to USPS formatting standards and verifying deliverability before print, automating ingestion in Python for batches of up to 250,000 addresses, where client-typical 8–15% defect rates meant reprints, wasted postage, and delays cascading into other clients' jobs.
- Modeled reporting data dimensionally with star and snowflake schemas, surfaced through SQL and BI dashboards.

Freelance

Freelance Developer

Aug. 2022 – Present

- Engineered automated ETL workflows to ingest, normalize, and validate raw client data from disparate third-party sources, enforcing the strict data hygiene and schema consistency required for trusted downstream workflows.
- Delivered short-cycle contract engagements integrating client systems with industry-specific CRMs and third-party APIs, working against vendor platforms.

UPS

Infrastructure & Monitoring Analyst

Sep. 2018 – Aug. 2022

- Contributed to an on-premise to hybrid AWS migration, working hands-on with Lambda, API Gateway, S3, RDS, DynamoDB, SNS/SQS, and CloudWatch across backend infrastructure supporting facility operations.
- Developed incident response and monitoring for an 80,000-package sort, one of three daily sorts, setting CloudWatch thresholds against a 15–20 minute downtime target where faults drove missorts and service failures.
- Transformed post-mortem incident reports into dimensional OLAP models that isolated analytical queries from transactional traffic, surfacing the recurring faults behind extended downtime so they could be addressed before repeating.

PROJECTS

retrain-pipeline

Training and governance

- CI-driven MLOps retraining pipeline on Amazon SageMaker in Go and Terraform at \$0/month idle cost: Great Expectations gates labeled data at the pull request in 5s for code-only PRs against 56s for data PRs, DVC versions datasets in an S3 remote and carries their content hashes into the Model Registry, and models land as PendingManualApproval under a CI role that holds zero wildcard IAM actions and no approval permission, so promotion requires a person.

inference-gateway

Serving

- LLM inference gateway in Go adding the operational layer on top of Amazon Bedrock: Server-Sent Events (SSE) token streaming, per-key API authentication and rate limiting, and retries with backoff and jitter. Meters token cost per request and attributes it per API key in structured logs, with per-key token buckets enforcing a hard ~\$10.50/day spend ceiling in middleware rather than a post-hoc billing alarm. Cut worst-case Bedrock API calls per request 67% (9 → 3) by replacing the AWS SDK's nested default retryer with a single explicit transient-only loop. Deployed on Amazon ECS Fargate with Terraform IaC.

rag-api

Retrieval

- Retrieval Augmented Generation (RAG) service in Go over HTTP: document chunking, Amazon Bedrock Titan v2 embeddings, pgvector similarity search (vector database), and grounded generation with source citations. Multi-tier VPC in Terraform IaC, keyless CI/CD to Amazon ECR over GitHub OIDC, and a distroless container on Amazon ECS Fargate.

TECHNICAL SKILLS

Languages:	Go, Python, TypeScript, JavaScript, SQL, Bash
System Design:	Serverless, Event-Driven Architecture, Microservices, REST APIs, Decoupled Architectures, Least-Privilege IAM, Observability, FinOps/Cost Optimization, ETL Pipelines, Dimensional Modeling (Star/Snowflake)
Cloud & Infrastructure (AWS):	Lambda, API Gateway, ECS/Fargate, S3, DynamoDB, RDS/PostgreSQL, Step Functions, SNS, SQS, EventBridge, CloudFront, VPC, IAM, CloudWatch, X-Ray, Terraform (IaC), Docker, Linux
AI/MLOps:	Bedrock, SageMaker, Model Registry, AI Agents, Model Context Protocol (MCP), LangChain, RAG, Embeddings, Vector Databases, pgvector, scikit-learn, pandas, NumPy, CI/CD (GitHub Actions, OIDC), DVC, Great Expectations, data quality gates
Certifications:	AWS Certified Developer; AI Engineer for Data Scientists Associate (DataCamp); Data Engineer Associate (DataCamp); Data Science with AI Professional Certification (Codecademy)