

# Amin Marashi

## Senior Software Engineer

Amsterdam, Netherlands · me@amin.codes · +31 6 5740 8572

[amin.codes](#) · [linkedin.com/in/amin-codes](#) · [github.com/aminmarashi](#)

Authorized to work in the EU without sponsorship.

## SUMMARY

Senior software engineer with 10+ years building backend and platform systems, focused on distributed, serverless architectures on AWS. I design decoupled, message-driven services that hold up as traffic grows, build the SDKs and tooling other engineers ship on, and treat scalability and security as design concerns rather than afterthoughts. Hands-on with Rust for performance-sensitive backends and Rust/WebAssembly frontends, and with the low-latency demands of real-time trading. AWS Certified Solutions Architect – Professional.

## EXPERIENCE

### Senior Software Engineer · New10 (ABN AMRO)

Oct 2022 – Present

*Amsterdam, Netherlands*

- Work across the Risk domain's serverless services on AWS (Lambda, API Gateway, SNS), built with the Serverless Framework and provisioned with Terraform and CloudFormation.
- Led the deprecation of a bulky core service, splitting it into two smaller, purpose-built ones I designed and rolled out across teams; one replaced the direct KVK (Dutch Chamber of Commerce) connection with an in-house ABN AMRO source, **cutting company-data lookup costs by more than half (€100K a year)**.
- Drove the architecture approval and cross-team buy-in to break a monolithic contract service into domain-specific microservices; moved contract generation onto Salesforce (Marketing Cloud) behind a backwards-compatible layer so old and new systems run in parallel during migration, then helped roll it out.
- Built platform tooling used across teams: added **beta publishing to the internal-package CI/CD pipeline** and a **company-wide log-anonymization process**.
- Improved developer experience: made a service runnable locally with serverless-offline and removed the bundler to cut build times.
- Designed integration and end-to-end testing tooling across services; hardened API-server middleware with stronger request/response validation. Hosted and presented the company-wide Service Engineering Meetings.

Stack: [TypeScript](#) [Node.js](#) [AWS Lambda](#) [API Gateway](#) [SNS](#) [Serverless Framework](#) [Terraform](#) [CloudFormation](#)

### Senior Software Engineer · Fashion Cloud

Jun 2021 – Oct 2022

*Amsterdam, Netherlands*

- Built and maintained an in-house data ETL tool that processed and merged high-volume product data in PostgreSQL, loading the merged result into MongoDB at the final stage.
- Led the re-architecture that fixed scalability, decoupling pipeline stages into independently deployed ECS containers and owning the rollout.
- Orchestrated async pipelines with AWS Step Functions and CDK; evaluated visual ETL tools for data transformation.

Stack: [TypeScript](#) [AWS](#) [ECS](#) [Step Functions](#) [CDK](#) [PostgreSQL](#) [MongoDB](#)

### Full Stack Engineer · Deriv.com (formerly Binary.com)

Jan 2016 – Dec 2019

*Cyberjaya, Malaysia*

- Designed and built [@deriv/deriv-api](#), a reactive SDK over the trading WebSocket API, plus its documentation, **used across all of the company's front-end apps**.
- Worked on a distributed, message-driven trading backend: thin WebSocket API layers delegating to RPC workers over Redis queues, decoupled and independently scalable, handling **sub-second price ticks and trades**.
- Proposed and built [Deriv Bot](#), a visual auto-trading platform (Google Blockly); maintained its [JavaScript-interpreter sandbox npm package](#).
- Wrote a Perl integration/load-testing framework (IO::Async, Future) and a streaming [WebSocket→S3 uploader](#) for KYC documents.

- Built the most successful front-end trading app on the WebSocket API (**generating millions in annual revenue**); briefly served as DBA (2019): DB hardening, replica routing, 5× faster integration tests.

Stack: [Perl](#) [JavaScript](#) [React](#) [WebSocket](#) [Redis](#) [PostgreSQL](#) [AWS](#)

**DevOps Tech Lead** · Deriv.com (formerly Binary.com)

Dec 2019 – Jun 2021

Cyberjaya, Malaysia

- Built a release manager on top of Chef and Slack **coordinating company-wide releases across hundreds of EC2 instances**; led a team of senior PostgreSQL engineers and ran round-the-clock on-call.
- Led the migration of an on-prem BI stack (incl. a 2 TB reporting database) to GCP with minutes of downtime, owning the architecture and cutover; cut GCP spend by thousands of USD/month.
- Automated tagging/release workflows by integrating Trello, GitHub, and Redmine; implemented database security measures and tested point-in-time-recovery (PITR) backups.

Stack: [PostgreSQL](#) [AWS](#) [GCP](#) [Docker Swarm](#) [Chef](#) [Airflow](#)

**Full Stack Engineer** · Freelance / Self-employed

Mar 2012 – Jan 2016

Remote

- Built a D3.js dashboard visualizing RADIUS authentication logs streamed via DynamoDB Streams and AWS Lambda (Eleven Wireless), an early event-driven serverless pipeline.
- Developed AngularJS dashboards on Django/Flask REST APIs, documented with Swagger; shipped cross-platform mobile apps in Python (Kivy).
- Implemented an IETF draft for IPv6 headers in the FreeBSD kernel (C); real-time radar dashboard for SkyRadar.

**System Administrator** · Shiraz University

Sep 2010 – May 2012

Shiraz, Iran

- Linux administrator for the CSE department; managed course-management and AD servers, and built a UI to orchestrate virtual PCs with importable snapshots.

## SELECTED PROJECTS

- **Serverless game-creation platform** (personal product): a distributed, serverless system on AWS. Published games run behind API Gateway with **Rust Lambda** backends and **DynamoDB** storage, frontends served from Cloudflare Pages. Decoupled, independently scalable services with security and scalability as first-class concerns. Backends written in Rust; frontends in **Rust compiled to WebAssembly** plus vanilla JavaScript.
- **Lit Feed** ([feed.lit.codes](#)), a distributed feed-reader: a Vercel full-stack frontend over a MongoDB Atlas worker system that refreshes feeds asynchronously, and a separate Python engine that classifies articles async.
- **ai-tools**, CLI utilities for AI-assisted development, including *cerebro*, an orchestrator that runs and monitors multiple Claude agents until they produce a reviewed, merge-ready pull request.

## EDUCATION & CERTIFICATIONS

### Education

**B.Sc. Computer Engineering**, Shiraz University (2009–2014)

**Cloud DevOps Engineer Nanodegree**, Udacity (2020)

### Certifications

AWS Certified Solutions Architect – Professional (2025)

AWS Certified Solutions Architect – Associate (2024)

Domain-Driven Design (DDD), O'Reilly (2024)

AWS Certified Developer – Associate (2022)

## SKILLS

### Languages

TypeScript, JavaScript, Rust, Python, Perl, C, SQL, Bash

### Distributed

Message-driven / event-driven architecture, decoupled services, async processing, WebSocket & REST APIs, API & SDK design, low-latency / time-sensitive systems

### Serverless / Cloud

AWS (Solutions Architect – Professional): Lambda, API Gateway, DynamoDB, SNS, Step Functions, ECS/Fargate, CDK · Serverless Framework · GCP

### Data

PostgreSQL, MongoDB, Redis, DynamoDB (key-value), ETL / data pipelines, DDD

### Practices

Infrastructure as code (Terraform, CloudFormation, CDK), CI/CD, integration & E2E testing, WebAssembly, observability, on-call/SRE

### Languages (human)

English (professional), Persian (native), Dutch (B2)