

# Raul de Hevia *Software Engineer*

🔗 [www.rauldehevia.com](http://www.rauldehevia.com)   🌐 <https://github.com/RdeHevia>   ✉ [raul.dehevia@gmail.com](mailto:raul.dehevia@gmail.com)   📍 San Francisco, CA

## PROFESSIONAL EXPERIENCE

---

### Co-Creator, Software Engineer

2021 | San Francisco, CA

*Monsoon* ([monsoon-load-testing.github.io](https://monsoon-load-testing.github.io)) 🌐

Monsoon is an open-source, serverless framework for running browser-based load tests in the cloud.

- Architected and constructed an auto-scaling, fully serverless load-generation engine using AWS ECS, Fargate, EventBridge and Lambda to spin up and control up to 20,000 Google Chrome instances, one per virtual user.
- Architected and built a serverless ETL pipeline using AWS EventBridge, Lambda and S3 to process and aggregate dozens of Gigabytes of raw test results before storing them in AWS Timestream (serverless time-series database).
- Use AWS EventBridge and Lambda to orchestrate the components of the ETL pipeline to allow auto-scaling and to process raw test data in near real-time.
- Prioritize scalability, fault tolerance, separation of concerns and ease of use when engineering solutions.
- Wrote a npm library (*monsoon-weather-station* 🌐) in Typescript to abstract away the complexity of writing the load-testing scripts as well as to automate the extraction of relevant performance metrics from Chrome instances using the npm library Puppeteer and the Performance Web API.
- Built a CLI wizard using AWS CDK and SDK libraries to automate the deployment of Monsoon's infrastructure and to allow users to create, configure and launch their tests.
- Developed Monsoon's visualization dashboard, Weather Channel, to allow Monsoon's users greater insight into the performance of their web app in Near Real Time using React, VictoryChart and Express.
- Authored comprehensive technical case study, readable at [monsoon-load-testing.github.io/case-study](https://monsoon-load-testing.github.io/case-study) 🌐

### Full-stack Developer

2020 – 2021 | San Francisco, CA

*Self-Employed*

- RequestChest: A Request Bin-style application to collect and inspect HTTP and webhook requests. Built with Node.js, an Express RESTful API, React, React Router and MongoDB and deployed to a DO Droplet via SSH.
- Cello: Project management app using an Express RESTful API, React, Redux and MongoDB.
- Flight routes dashboard: Full stack web app for viewing the available routes between two airports built with Express, React, Redux, containerized with Docker and deployed in AWS with Elastic Beanstalk.

### Structural Engineer

2014 – 2020

*Arup* 🌐

San Francisco (CA), UK, Spain

- Worked in a wide array of international projects, regularly interfacing with other engineering teams, clients and public agencies.
- Created tools and workflows in VB to process raw data from structural models and obtain different insights.

## TECHNOLOGIES

---

### Languages

JavaScript, Typescript, Go, SQL, HTML, CSS, VB, C

### Frameworks

Express, React, Redux, TailwindCSS

### Misc. Technologies and Skills

Node.js, PostgreSQL, MongoDB, Docker, Puppeteer, Jest, Git, Github, Postman, HTTP, OOP, RESTful API, Numerical and Statistical Methods

### AWS

Lambda, ECS, Fargate, S3, Timestream, EventBridge, VPC, CloudWatch, CloudFormation, IAM, SDK, CDK

## EDUCATION

---

### Software Engineering & Full-stack Web Development

2019 – 2021

*Launch School* 🌐

### BsC and MsC in Civil Engineering

2008 – 2015 | Spain

*Technical University Of Madrid* 🌐

*Exchange student in Delft University of Technology* 🌐, Netherlands (2012)