Matt Femia | Software Engineer

New York, NY
github.com/mattfemia
linkedin.com/in/mattfemia

TECHNICAL SKILLS

Strong: Python, JavaScript (ES6+), Flask, TypeScript, Node.js, React.js (Hooks & Router), Express.js, SQL (MySQL, PostgreSQL), Redis, Webpack, version control (Git/Github), Docker, OOP, AWS (EC2, ECS, RDS, Lambda, ALB, IAM, VPC, S3, Cloudwatch, Cloudfront, API Gateway, SQS, SNS, Route53, Elasticache), PyData (pandas, numpy, matplotlib, scipy), HTML, CSS/SASS, Bash, Agile/Scrum (JIRA)
Experienced: Travis CI, Terraform, Nextflow, Redux, NoSQL (MongoDB), C/C++, testing (PyTest, Jest, Supertest)

EXPERIENCE

Accumulus | Full-Stack Software Engineer | Open-source AWS Lambda monitoring and cost analysis application **2022 - Present**

- Leveraged AWS SDK for Node.js to implement AWS Lambda metric monitoring and parsing of Cloudwatch log data to deliver actionable insights to end users, providing clarity towards reducing overhead AWS expenses and optimize Lambda functions
- Harnessed AWS Lambda and EventBridge to generate sample user data and site traffic from custom Python functions to load-test application in production environment and ensure consistent performance across several KPIs to optimize UX
- Defined Typescript interfaces and custom types to enforce static type error handling for aggregated data objects throughout the application which guaranteed scalability, minimized errors, and improved maintainability across the codebase
- Utilized AWS ECS, ECR, Route 53, Cloudfront, Cloudformation, VPC, ASGs, ALB, and S3 to implement application architecture, deployment, DNS configuration, security, and asset management, providing a secure, scalable, and highly-available application
- Used React graphing libraries to construct dashboard visuals for Cloudwatch metric and log data to abstract away the need for manually parsing through static files, leading to accelerated debugging and monitoring for developers and devops teams
- Implemented Agile methodology / Scrum workflow using Jira Board to plan sprints, issue tasks, and manage project progress
- Configured CI/CD test and deployment pipeline orchestrated with TravisCI to execute test suite prior to AWS EC2 deployment
- Product developed under tech accelerator OS Labs (opensourcelabs.io)

The Micrographer, LLC | Owner, Manager | eCommerce business offering fine art microscopy prints **2020 - Present**

- Utilized Python with Flask web framework to develop an e-commerce application, leading to 20X increase in monthly sales
- Used React with hooks for internal single-page applications to create modular frontend UIs with streamlined state management
- Leveraged Docker with AWS to design a fault-tolerant and scalable architecture due to ease of container management with AWS ECS clusters and EC2 auto-scaling customization, achieving >99.999% uptime YTD
- Incorporated AWS Cloudfront CDN, webp image compression, caching, and minification strategies to increase page-speed by 95%, allowing customers to have improved UX and ultimately decrease bounce-rate
- Configured AWS SQS, SNS, and Lambda to decouple expensive compute processes from core eCommerce site by using the publish/subscribe design pattern to create asynchronous message queuing, resulting in fast UI updates to improve UX

Memorial Sloan Kettering Cancer Center | Research Technician | Immuno-oncology research for breast cancer **2018 - 2020**

- Used R and Python to conduct multivariate analysis of genomic data by leveraging mature bioinformatics and data science libraries, (i.e. pandas, numpy, matplotlib, tidyverse), leading to published research discoveries in leading scientific journals
- Harnessed Python (Flask), React, and AWS to build multiple web applications for automating team's data analysis, tracking inventory, and optimizing scheduling allowing the research team to focus more on assays and accelerate project lifecycle

OPEN-SOURCE

scRNA-Seq Analysis Pipeline | Genomic data analysis pipeline orchestrated with Nextflow and Terraform

- Utilized Nextflow to orchestrate data workflow by integrating Bash and Python analysis files into scalable pipeline configuration
- Used Bash scripts to handle argument-parsing of pipeline options from user-inputs, offering versatile customization of analysis
- Harnessed Terraform to automate environment lifecycle and provide end-user seamless access to AWS resources

Clonotypy | Python package for reproducible multivariate analysis and visualization of immuno-sequencing data

- Leveraged Python STL to accelerate file I/O and decrease computational bottlenecks with multithreading & multiprocessing
- Used Python's requests module to develop several built-in SDKs for medical/scientific REST APIs, providing developers the convenience of querying data against an aggregated collection of databases and generating instant data reports
- Utilized matplotlib and seaborn to construct data visualizations based on research publication aesthetics providing users with instant publication-quality visuals, while also providing flexibility to export JSON configs for rendering JavaScript-based graphics

EDUCATION

Biology with Thesis Option, M.S. | College of William & Mary
Exercise Sciences with Medical Emphasis, B.S | Ithaca College

PUBLIC TALKS

Serverless Architecture with AWS Lambda: Single Sprout Software Engineer Speaker Series | 2022

CERTIFICATIONS

AWS Certified Solutions Architect - Associate (SAA)