Ethan S. Rabb

ethanrabb@gmail.com | (440) 221-9329

Education

MS/BS in Computer Science; minor in Robotics — *Washington University in St. Louis,* 2009-2014

Skills

Languages: Java, Python, Javascript, C/C++/C#, Ruby, HTML, CSS

Skills: Full Stack Development, Cloud/Microservice Architecture, Big Data ETL, Distributed Systems, High throughput / low-latency networking, Production testing/deployment

Technologies: Spring, React, Django, REST, Webpack, Typescript, JPA, PostGreSQL, Docker, AWS, Git, Linux, Bash, Docker, Flyway, jQuery, A-Frame, WebGL, GLSL, ROS, Gazebo, Node.js, Maven, TeamCity, Jenkins, IntelliJ/PyCharm, Puppet, Chef, Terraform, Eclipse

Work Experience

Senior Software Engineer, *Imprivata/SecureLink/Maize Analytics*
June 2021-*Present*

Led full-stack development on a healthcare data privacy auditing website and storage system

- Improved and maintained website serving 50+ hospitals distributed across the US, and ingesting millions of records/day
- Advocated for and led planning and execution of a full-scale website transition from JavaScript to React
- Created and refactored backend systems in Python/Django, and frontend systems with JavaScript/HTML/CSS
- Designed and optimized queries and schema in PostgreSQL for a complex, proprietary DB structure
- Worked with Product and Design teams to architect new features according to spec and client needs
- Worked with Operations and Support teams to improve cloud architecture, security and authentication, and internal tooling

Server Developer, *Foreflight (Boeing)* April 2018-January 2021

Managed weather and notification infrastructure for two apps used by a majority of the world's pilots.

- Development primarily with Java/Spring and Python/Bash, running on Terraform-defined AWS services.
- Ownership of multiple services used for ETL on a diverse array of graphical and text data streams.
- Designed and Implemented a large-scale modular redesign of weather systems, using load-balanced microservices and contract API's
- Created a variety of internal tools: Web interfaces, monitoring systems, and persistent bootapps
- Worked with logbook team to add features and fix bugs with customer-facing single-page web interface in React

Software Engineer, **Edge Services** *Quantcast* June 2014-August 2017

Managed a large-scale distributed system providing high-speed infrastructure for real-time bidding.

- Handled millions of requests/second with 2 dozen services, on 1000+ physical & cloud hosts in 15+ datacenters.
- Development in Java, C++, Python, Bash, Docker, and more, including proprietary systems like qSQL and QFS
- Ownership of multiple services (feature addition, bug fixing, unit/integ-testing, platform migration, deployment)
- System design, performance optimization, operational excellence (scripting and on-call), cloud migration planning

Research Intern, *University of Arizona CATVehicle REU* June 2013-August 2013

- Designed an interface between ROS and JAUS, using a custom node structure written in C++.
- Collected data and sent commands to and from a JAUS-based autonomous vehicle using ROS.

Control Systems Intern, *Intelligrated, Inc* June 2012-September 2012

- Created, edited, and tested movement and mapping simulations with Gazebo and Webots.

Projects

Built an immersive, cross-platform 3D VR music video using Javascript and A-Frame. Created custom shaders, procedural asset generation, state management, message-based synchronization, and more.

Created a dynamic, procedural VR visualizer using fractal math with GLSL. Combined with an A-Frame app to allow users to control almost a dozen custom variations for near-infinite visual possibility.

Designed and implemented an inline filtering solution for upload files at scale, providing significant performance optimizations and intelligent, configurable failure handling tailored to the needs of support teams.

Created an interactive command-line utility for automated stock trading, complete with buy/sell signal detection, profit tracking, back-testing, and an A/B testing framework.

Wrote a long-running Spring Bootapp for polling and storing data from two aggregation API's at scale. Used parallel Java streaming to parse and collect data, handled app-specific authentication, and worked with customers to build appropriate DB schema.

Used Javascript and Spring to create a dynamic web interface, allowing teammates multi-filter search and CRUD access to restricted DB.

Implemented a variety of custom Java API endpoints, polling, parsing, and transforming data to customer specifications

Migrated and owned several Spring Boot apps, updating and improving on relevant Quartz jobs, APIs, authentication, persistence, business logic, etc.

Created a modular API contract monitoring service, redesigning monitoring dashboards and adding automated systems to simplify alerting responses and significantly reduce operational load.

Designed a database-backed customizable notification system, allowing internal customers to create regex-matching alerts with personalized polling jobs

Built a live log file duplication and comparison infrastructure for production deployment testing, with 100% uptime during deployment.

Owned high throughput Java http server communicating with multiple large distributed systems, and created custom testing and deployment procedures to handle throughput and production transfer to upgraded versions.

Added features and environments to a generalized C++ testing framework (Gtest integrated) used for input and load testing.

Designed and automated production canary tests for many new service features, allowing for upgrades to systems with strict always-on requirements.

Created, tested, and deployed new features for large-scale file download service run in Java, including file prioritization and ordering, peer-to-peer networking, status endpoints, increased connection stability, and error handling.

Wrote and improved Python and Ruby scripts for a variety of operational needs, like log file prioritization, simplification of deployment procedures, dynamically setting the IP of new AWS instances, etc.

Used Jenkins, Terraform, and Puppet to automate an auto-scaling group of instances running a high-throughput log collection service, allowing for quick update deployment at all levels of the pipeline.