Jacob Smith

Greenville, SC jacob@smithjs.org linkedin.com/in/jacobthemyth github.com/jacobthemyth

Professional Summary

Platform Engineer / Site Reliability Engineer with extensive experience optimizing performance-critical applications and implementing infrastructure-as-code and CI/CD pipelines. Motivated by developing intuitive abstractions that allow developers to work effectively with complex systems.

Languages and technologies

- **Languages**: Ruby, Go, Bash, Lua (for Redis and nginx), SQL, C, JavaScript (Node.js), Python
- **Distributed Systems & Databases**: PostgreSQL, Redis, Kafka, Cassandra, Elasticsearch, S3, DynamoDB
- **DevOps**: AWS, GCP, Kubernetes, ArgoCD, GitHub Actions, Terraform, Kustomize

Work Experience

DevOps Consultant **Freelance (Remote)** **October 2023 - Present**

- Refactored an untested and brittle C/Ruby integration based on SWIG, enabling Ruby developers to make API changes without C knowledge while maintaining runtime safety and performance
- Resolved a catastrophic outage by identifying and resolving virtual machine I/O bottlenecks, optimizing workload distribution across virtual block devices to account for network latency overhead
- Authored detailed runbooks and automated workflows to ensure long-term system resiliency

Staff Engineer **Kajabi (Remote)** **July 2021 - July 2023**

- Provided technical leadership for 30+ staff across Production Engineering, UX, Quality Engineering, and Security & Risk
- Guided 9 cross-functional teams in data systems design and implementation, maintainable abstractions, and performance
- Mentored engineers in application performance and designing resilient interfaces to distributed systems like Kafka and DynamoDB
- Designed content-addressable storage for rendering end user templates, significantly reducing memory usage across processes

Staff Production Engineer **Kajabi (Remote)** **December 2020 - July 2021**

- Rebuilt the CI/CD pipeline for a large Rails application using Docker and BuildKit, reducing median build time from 45 to 9 minutes through careful optimization of layer caching, dependency caching, conditional step execution, and parallel execution and as a side effect, allowed builds to be run locally with fully reproducible build artifacts.
- Optimized Aurora PostgreSQL to handle 50,000 QPS during peak loads through configuration tuning and query optimization
- Implemented database performance isolation by decoupling workloads with a foreign data wrapper (FDW), preserving functionality while preventing system-wide slowdowns

Senior Production Engineer (Tech Lead) **Kajabi (Remote)** **November 2019 - December 2020**

- Defined the hiring and onboarding processes for the Production Engineering team and grew the team from 2 to 7 engineers while managing 2 full time contractors.
- Designed and executed a near-zero downtime migration of a Heroku PostgreSQL database to AWS Aurora PostgreSQL, leveraging kernel tuning and cache warming to reduce customer impact and proactive testing and rollback planning to ensure resilience.
- Developed infrastructure-as-code, migration automations, and provided technical leadership to the Production Engineering team and the 6 product development teams for migrating application compute workloads from Heroku to AWS EKS.
- Implemented libraries and services in Ruby, primarily around resilience in the face of partial system failure

Senior Software Consultant **Subcontractor for Test Double (Remote)** **May 2019 - November 2019**

- Worked on the Ruby VM (i.e. “MRI”) in C to prototype the reliability and performance impact of interning all string literals.
- Diagnosed and fixed memory problems in a Go service that used `cgo` to integrate with a Rust library that itself integrated with a C++ library, reducing memory usage of ~10k server instances by ~250MB each, significantly decreasing resource utilization across a Kubernetes cluster