David Ly

510-828-9675 | david.ly@berkeley.edu | davidmacly.dev | david-ly | david-mac-ly

Experience

LogDNA n.k.a. Mezmo | *Software Engineer II*

Jan 2022 - Jun 2022 | Mountain View, CA

- Developed as an individual contributor with oversight on key parts of the codebase (wrt backend functionality primarily)
- Handled code review prioritizing areas of expertise with greater discretion in ensuring solutions focused on: maintainability, implementation efficacy, alignment with best practices, consistent style/linting, & standardized usage patterns

Data Pipeline Efforts [Node.js + Jira]

- Drafted an agile epic to scope out the necessary steps to instrument a durable queue at each stage of the data flow
- Tackled the refactor of the notification microservice (send function) for integrations with OpsGenie, PagerDuty, etc

Provision API [Node.js]

- Aided in building a fully fledged provisioning API, spurred by & intended for enterprise use, which aligned with simultaneous RBAC work to scaffold a hierarchical management structure supported with inheritable & reusable components

Terraform Plugin [Node.js + Go(lang)]

- Worked on our custom Terraform provider to align its resources towards consistency with the views & alerts API schema
- Revamped a large portion of the TF plugin's codebase for better CRUD parity & added support for Slack alerts integration

LogDNA n.k.a. Mezmo | *Software Engineer I*

Jan 2020 - Dec 2021 | Mountain View, CA

- Developed as an individual contributor with light familiarity across the entire (mostly) codebase given prior internships
- Engaged in code review with strategic focus on engagement conducive to broader learning & understanding of concepts

On-call Proceduralization [Node.js + CLIs (bash, kubectl, etc)]

- Fulfilled on-call developer duties via debugging culpable code, deploying hot-fixes, rolling back faulty deploys, etc
- Pushed for an new on-call rotation system to more evenly distribute the workload across the entire development team

OpenAPI Standardization [Node.js + Swagger]

- Advocated for the adoption of best practices for RESTful APIs & standardization across development & documentation
- Initiated the instrumentation of Swagger(hub) tooling to better align our documentation with the OpenAPI specification

Elasticsearch Migration & Upgrade [Node.js]

- Heavily contributed to efforts involving the migration & upgrade of our system(s) from Elasticsearch versions: 5.x → 7.x
- Refactored numerous jobs & workers, tailored for Elasticsearch, updating much of their core logic to facilitate the transition from tribe nodes to cross-cluster search & account for changes in their Query DSL (Domain Specific Language)

Export API v2 [Node.js]

- Led development + documentation processes for the feature release of our v2 API letting users export their logs as JSON
- Leveraged the search_after parameter of Elasticsearch's API to offer native support for deep pagination on query results

LogDNA n.k.a. Mezmo | *Software Engineering Intern*

May 2019 - Aug 2019 | Mountain View, CA

- Targeted monitoring & observability into backend systems by instrumenting Prometheus clients to emit OpenMetrics
- Implemented & modularized archiving functionality for various storage providers (AWS S3, Azure Blob, IBM-COS, etc)
- Initiated the introduction of AWS Marketplace SaaS Subscription Services into our various integrated product offerings

Skills

Programming Languages: Javascript (+ Node.js), Shell (Bash), Golang, HTML/CSS, YAML, Java, Python, C(#), SQL

Technical Background: Git, Unix, Kubernetes, Docker, MongoDB, Redis, Elasticsearch (& Lucene), Swagger (& OpenAPI), Prometheus (& OpenMetrics), Kafka, RabbitMQ, Hashicorp Configuration Language (HCL), Terraform, Jenkins, CircleCI

Education

University of California, Berkeley

Berkeley, California

B.A. Cognitive Science (Computational Modeling Focus) | Minor in Computer Science — Aug 2015 - Dec 2019

Relevant Coursework: Data Structures, Computer Architecture, Operating Systems, Database Systems, Computer Security, Artificial Intelligence, Internet Architecture & Protocols, Efficient Algorithms & Intractable Problems