https://github.com/evanlucas
https://evanlucas.com
evanlucas@me.com

Evan Lucas

Node.js core contributor and expert

Education

B.S.B.A. Finance from the University of Southern Mississippi (December 2010)

Employment

Mezmo (Formerly LogDNA) - *Principal Software Engineer* - (Jan 2021 - Present)

Mezmo (Formerly LogDNA) - *Senior Software Engineer* - (Apr 2020 - Jan 2021)

- Drive architecture improvements across the backend
- Drive best practices for using Node.js in production
  - Write blog posts on debugging memory/cpu/event loop/thread pool issues in Node.js services
- Lead of special projects team that focuses on high priority issues
  - Started cost reduction project in AWS for cross-AZ network traffic
    - saved millions per year 2 weeks (~75% reduction)
  - Started cost reduction project for using arm64 in AWS (could net ~$20k/month savings)
  - Significantly reduced data loss/corruption in message broker
    - Saved over 1 Tebibyte of memory with this rewrite
  - Reduced disk I/O by 40%, memory usage by 70% in message broker
- Maintain tooling that makes switching between various kubernetes clusters significantly easier
- Built out a lot of internal monitoring that allowed us to alert on issues before customers
- Built various tooling to make the deployment process easier
- Drove change for writing automated test suites for all projects
- Wrote automatic versioning tool that handle bumping versions for packages/services in CI.
- Maintained docker images for development and production services

Help.com - *Chief Technology Officer* - (Mar 2018 - Mar 2020)

- Managed multiple engineering teams (both remote and onsite)
- Contributed heavily to both frontend and backend systems (React, Node.js, Golang)
- Designed backend infrastructure on Kubernetes including operator to make deploying services easier
- Significantly enhanced deployment process with GitLab and Kubernetes
- Improved delivery process by creating review apps for frontend and backend teams
  - Developers and QA were able to test changes in isolated environments before merging code

Help.com - *Principal Architect* - (Oct 2017 - Mar 2018)

- Designed backend systems
- Ensured code quality
- Identify and correct low performing services/endpoints
- Maintained docker images for development and production services

Help.com - *Senior Software Engineer* - (Nov 2015 - Oct 2017)

- Built a lot of the building blocks for full system rewrite
- Wrote high performance microservice library
  - Features code generation for validators and data structures
  - Has built in tracing for use with Stackdriver
- Wrote high performance API Gateway for our infrastructure

Help.com - *Software Engineer* - (Jun 2015 - Nov 2015)

- Contributed to both frontend and backend application development
- Significantly improved performance of prototype application
  - Rewrote client communication library to boost performance from 500 messages total to 100k messages/sec
- Introduced testing strategy for backend microservices

GAW - *Software Engineer* - (May 2014 - Mar 2015)

- Contributed to Node.js application that used MongoDB and redis

Hattiesburg Clinic, PA - *Web Developer* - (Oct 2012 - May 2014)

- Built various Node.js applications

Technical Skills

- **Languages**: JavaScript, Golang, Objective-C, HTML, CSS, Swift, Tcl, C, C++, PHP
- **Databases**: Postgres, Elasticsearch, MongoDB, MySQL, MariaDB, MS SQL Server, Redis, Couchbase
- **Monitoring**: Prometheus, Grafana, Sysdig
- **Shells**: fish, bash
- **Orchestration/tooling**: docker, kubernetes, helm, razee
- **CI/CD**: Jenkins, Gitlab, GitHub Actions

Some examples of codebases I have written/worked on

- **ingest**: A HTTP ingestion service for logs that handled upwards of 200k requests/s.
  I made significant performance improvements to this service (JavaScript, Rust)
- **cat-files**: A native addon for Node.js that concatenates multiple files into one significantly faster than using built in solutions (streams, etc). The goal was to improve performance while maintaining memory pressure. This improved performance by about 30%. (C, C++, JavaScript)
- **napi-lz4**: Internal replacement for the `lz4` on npm with ~200x performance improvement (Rust, JavaScript)
- **releaseit**: CLI tool to automate releasing packages and services.
  Uses commit metadata to determine versioning, bumps, tags, and builds release artifacts. (JavaScript, docker)
- **docker-runner**: A project similar to docker-compose that allowed us to run test suites with docker containers easier (JavaScript)

Open Source

- **Node.js**: Technical Steering Committee Emeritus and former member of the Release team
  - Helped port Node.js over to arm64 when Apple Silicon came out
  - Added ability to generate code coverage for Node.js core on macOS.
  - Added `crypto.randomFill{Sync}()` as a faster alternative to `crypto.randomBytes()`
  - Added the `lookup` functionality to `Socket#connect`
  - Improved error metadata by adding the hostname/ip and port on network errors
- **fish-kubectl-completions**: fish shell completions for the kubectl CLI tool (Golang, fish)
- **core-validate-commit**: CLI tool to validate commit messages for Node.js core
- **build-ast**: Node.js package to make generating source code via AST easier
- **gcr**: A GitLab CI Runner for node.js (JavaScript)

Links

- **Node.js** (https://github.com/nodejs/node)
- **fish-kubectl-completions** (https://github.com/evanlucas/fish-kubectl-completions)
- **core-validate-commit** (https://github.com/nodejs/core-validate-commit)
- **build-ast** (https://github.com/helpdotcom/build-ast)