

Mykolas Viningas

📍 London, UK 📩 m.viningas@gmail.com 🔗 mykolasviningas.github.io 💬 linkedin.com/in/mykolas-viningas

PROFESSIONAL EXPERIENCE

Smartsheet ↗

An enterprise work management platform

London, UK/remote

Technologies used: Go, AWS, Terraform, SQL, NoSQL, Kubernetes, Linux, CI/CD, Datadog

Software Engineer II - 11/2024 - present

- Improved service reliability and proactive issue detection by implementing *User Canaries* that continuously validate event pipelines; replaced a reactive “flying blind” approach with automated monitoring and metrics.
- Boosted developer efficiency and reduced testing friction by creating the *Request Bin* internal tool, enabling engineers to self-serve webhook endpoints for testing and debugging—now frequently used and integrated into automated tests.
- Elevated API performance and reliability to meet the 99.95% SLO by diagnosing and fixing critical Lambda memory and SQL query inefficiencies, cutting latency, and fixing failures in production workloads.
- Enhanced customer experience and minimized downtime by implementing retry logic for external service calls, aligning the API reliability with industry standards, and maintaining SLO compliance.
- Enabled scalability and future growth by designing and implementing a distributed BI event reporting solution capable of processing more than 10x more data, unblocking the API migration, and supporting revenue-linked analytics.
- Supported junior engineers by unblocking technical challenges, knowledge sharing, and providing feedback on their work.

Software Engineer I - 08/2023 - 10/2024

- Enabled new async event capabilities in production by setting up the required infrastructure and pipelines, unblocking multiple internal consumers dependent on those events.
- Strengthened observability and incident responsiveness by replacing delayed metric-based alerts with direct CloudWatch triggers to PagerDuty, enabling immediate visibility into event queue delays.
- Contributed to multiple technical design reviews (authorization, data migration, and API migration) and consistently advocated for scalable, cloud-native architecture aligned with customer needs.
- Supported production reliability by participating in on-call rotations, diagnosing and resolving incidents, and contributing to post-incident reviews to prevent recurrence.
- Partnered with support and platform teams to define operational maintenance tooling for post-migration service, clarifying requirements and proposing forward-looking solutions that reduced dependency on legacy systems.

Symphony (symphony-framework.github.io) ↗

An open-source real-time collaboration framework

01/2023 – 07/2023 | remote

Creator, Software Engineer

- Researched and decided on the components to be provisioned, including real-time infrastructure and state convergence.
- Implemented preservation of shared application state (document) by checkpointing and storing the documents in binary form to object storage (AWS S3).
- Designed the schema for a PostgreSQL database (AWS RDS) and implemented persistence of server metadata.
- Simplified the monitoring and debugging of the system by building a dashboard interface that provides metric data about connections and documents (React, TypeScript, Tailwind).
- Wrote a RESTful API (Node.js/Express) for delivering server metadata to the dashboard client, including live metrics (SSE).
- Abstracted away the backend management by containerizing (Docker, AWS ECR) and moving the architecture from AWS EC2 to ECS.
- Separated concerns by decoupling the backend into individual services for Dashboard and WebSocket clients.
- Provided a single point of entry to different services and traffic routing to individual instances of the WebSocket server by configuring an application load balancer (AWS ALB).
- Increased the number of concurrently connected clients handled by the WebSocket service from 240 to 10,000 (40x) by horizontally scaling the WebSocket server using pub/sub model (AWS ElastiCache for Redis) and allowing server-to-server querying for application state (AWS DynamoDB)
- Authored Symphony's case study: symphony-framework.github.io
- Collaborated as part of a remote team of 4 engineers across 2 timezones using Agile workflow

SKILLS

Back end

Go, Node.js, Express, SQL, NoSQL, RESTful APIs

Front end

JavaScript, TypeScript, React, jQuery, HTML, CSS

Tooling

Git/GitHub, GitLab, Docker, Linux, Kubernetes (EKS), Bash, Nginx

Cloud

AWS, Terraform, DigitalOcean Droplets, Heroku

EDUCATION

Launch School ↗

A mastery-based study of Software Engineering fundamentals

2019 – 2022

Royal Academy of Music, University of London

Bachelor's degree, Jazz/Jazz studies

2013 – 2017