Eric Lam

Carlton, VIC, Australia | ericlamwork15@gmail.com | +61 494 308 654

Summery

Experienced Full-Stack Developer with a strong background in building scalable, high-performance web applications. Proficient in **Go, Python** and **Database Techniques** for backend development, with expertise in **Kotlin** for modern, efficient coding. Skilled in frontend frameworks, including **React, Angular, and Vue**, to create dynamic, user-friendly interfaces. Hands-on experience with **AWS cloud services**, ensuring robust, secure, and scalable application deployment. Passionate about writing clean, maintainable code and optimizing system performance for enterprise-grade applications.

Education

Victoria University

Bachelor's Degree in Computer Science
Mar 2008 – Apr 2012

Technical Skills

- **Programming Languages:** Go, JavaScript (ES6+), TypeScript (4.x), Python (3.x), C#.
- **Backend Technologies:** Spring Framework (Spring Boot 2.x/3.x, Spring MVC, Spring Security, Spring Data JPA, Spring AOP, Spring DAO, Spring context), Microservices Architecture, RESTful APIs, GraphQL, JPA/Hibernate (5.x/6.x), Django, FastAPI, .NET, Node.js, Express, Nest.js
- **Frontend Technologies:** React.js (17/18), Redux, Mobx, Flux, RemixJS(2.8), Next.js, Angular (12/16), Vue.js (2.x/3.x), HTML5, CSS3, SCSS, Tailwind CSS (3.x), RxJS (7.x), Material UI, jQuery, Vuex, Nuxt
- **Cloud & DevOps:** AWS (EC2, S3, Lambda, RDS, DynamoDB, API Gateway, CloudFormation), Docker (24.x), Kubernetes (1.27+)
- **Version Control & Collaboration:** CI/CD (Jenkins, GitHub Actions, GitLab CI/CD)
- **Messaging:** Apache Kafka(3.x), Rabbit MQ(3.x), Active MQ(5.x), gRPC, SNS, SQS
- **Database Management:** PostgreSQL (14.x/15.x), MySQL (8.x), MongoDB (5.x/6.x), Redis (7.x)
- **Testing & Quality Assurance:** JUnit (5.x), Mockito (4.x), Jest (29.x), Cypress (12.x)
- **Version Control & Collaboration:** Git (2.40+), GitHub, GitLab, Bitbucket, Agile (Scrum, Kanban)

Role highlights

- **Led** the **design** and **implementation** of scalable systems, improving platform reliability and ensuring high availability during peak traffic periods.
- Spearheaded the **migration** of **legacy systems** to microservices, reducing downtime and enhancing system scalability.
- **Directed** the **architecture and development** of high-performance APIs, enabling seamless integration with internal and third-party services.
- **Mentored and trained junior developers**, fostering a culture of collaboration, knowledge sharing, and technical growth within the team.
- **Coordinated cross-functional efforts**, collaborating with product managers, UX designers, and DevOps teams to ensure business objectives were met.
- **Drove the successful implementation** of real-time data processing solutions, improving response times and reducing backend load.
- **Played a key role** in improving platform security, implementing robust authentication and authorization strategies.
- **Led the development of automated testing strategies**, reducing production defects and ensuring consistent quality across releases.
- **Managed end-to-end system migrations**, ensuring smooth transitions without data loss or service disruptions.
- **Advocated for and implemented** best practices in development, resulting in improved team productivity and faster feature delivery.
- **Led performance optimization initiatives**, identifying bottlenecks and improving system efficiency under heavy load.
- **Oversaw the successful deployment** of cloud-native solutions, ensuring scalability, cost efficiency, and high availability.
- **Drove innovation by implementing** new technologies and methodologies, improving system architecture and development processes.

PFOFESSIONAL EXPERIENCE

Senior Full Stack Developer , Codal - Chicago, Illnois, United States Sep 2021 – Nov 2024

- Developed high-performance **RESTful APIs and GraphQL endpoints** using **FastAPI (Python), Gin (Golang), and Next.js (React 18+)**, ensuring seamless integration with Codal's payment system, inventory management, and third-party services.
- Led the design and implementation of a scalable microservices architecture for Codal's Marketplace Order Processing System, utilizing **Python(3.11+, FastAPI 0.100+, Flask 2.3+)**, **Golang(1.20+, Gin1.9+, Fiber 2.50+)**, **React.js(18+), Next.js(14+), and Redux Toolkit(1.9+)**.
- Led the migration of legacy monolithic systems to a **microservices architecture using FastAPI (Python), Fiber (Golang), and Next.js (React 18+)**, improving scalability and reducing downtime by 30%.
- Built and optimized **React.js and Next.js frontends** for order management dashboards, improving user experience and system responsiveness.
- Integrated **Redux Toolkit for state management**, improving frontend performance and reducing unnecessary re-renders.
- Migrated database systems from **legacy SQL-based systems to Amazon DynamoDB and PostgreSQL (15+)**, ensuring zero downtime and improving data retrieval speeds.
- Spearheaded the migration to **AWS cloud infrastructure**, transitioning applications to **AWS EC2, S3, Lambda, and RDS**, increasing scalability and reducing operational costs by 20%.
- Designed cloud-native solutions on AWS, enabling the platform to **auto-scale using Kubernetes (1.27+), Docker (24.0+), and AWS Auto Scaling** to handle peak traffic periods like Prime Day and Black Friday.
- Increased platform security by implementing **OAuth2 and JWT authentication** using **Authlib (Python), OAuth2 package (Golang), and NextAuth.js (Next.js 14+)**.
- Led the development of an **event-driven architecture using Apache Kafka (3.5+) and RabbitMQ (3.11+)**, handling asynchronous communication for order processing, inventory updates, and customer notifications.
- Optimized data access and query performance using **SQLAlchemy (2.0+ in Python), GORM (v1.24+ in Golang), Prisma ORM (Next.js), and Redis (7.x)**, reducing database load by 40%.
- Implemented **AWS SQS** for managing order processing workflows and customer notifications, enabling real-time updates for order fulfillment and increasing processing speed by 20%.
- Designed and executed **CI/CD pipelines using Jenkins, GitHub Actions, and GitLab CI/CD**, reducing integration times and enabling faster feature delivery.
- Led the design of **high-availability solutions using AWS CloudFormation, Kubernetes (1.27+), and Docker Swarm**, ensuring services could scale horizontally while maintaining availability.
- Built interactive and high-performance **React.js** and **Next.js dashboards**, enhancing admin monitoring and analytics capabilities.
- Played a crucial role in designing **API contracts and implementing OAuth2 authentication** using **FastAPI's Security utilities (Python), Golang's OIDC provider, and NextAuth.js (Next.js 14+)**.
- Conducted extensive testing for data integrity during the migration of order management systems, including **unit testing, integration testing, and stress testing**.
- Led the implementation of **test-driven development (TDD) practices**, creating unit tests with **PyTest (Python 7.x), Testify (Golang v1.7+), and Jest (React 18+)**, reducing production defects by 25%.
- Developed and executed **automated regression tests** to ensure that newly deployed features did not impact existing system functionalities.
- Implemented **performance testing strategies using Locust (Python 2.15+), k6 (Golang v0.44+), Lighthouse (Next.js), and AWS CloudWatch**, optimizing microservices for high load.
- Led the development of **data migration tests**, ensuring seamless and reliable transfer of transaction data from legacy databases to DynamoDB.
- Supported **load testing efforts** for cloud-based services, ensuring **AWS Lambda and DynamoDB** could handle massive traffic loads without degradation.
- Integrated **unit tests and mocking frameworks into CI/CD pipelines**, reducing human errors and ensuring that only tested, validated code was deployed.
- Mentored junior developers on **writing unit tests in PyTest, Testify, and Jest**, improving team adherence to quality assurance processes.
- Led the implementation of **rollback strategies for failed migrations**, ensuring that services could revert to stable states without data loss.
- Developed **resilience patterns such as circuit breaking and retry mechanisms** using **Resilience4j (Python), Hystrix (Golang), and SWR (Next.js)**.

- Collaborated with the architecture team to create a **distributed transaction system with gRPC (Go-GRPC v1.55+, gRPCio 1.56+ in Python**, and **GraphQL in Next.js)**, reducing latency in inter-service communication.
- Implemented **real-time messaging solutions using AWS SNS** and **WebSockets in Next.js**, triggering customer notifications on order status, shipping updates, and promotions.
- Designed and executed **regression testing strategies**, ensuring that newly implemented features did not introduce regressions or impact system stability.
- Contributed to the successful implementation of **multi-region deployments in AWS**, ensuring **low-latency access to Amazon Marketplace services** globally while maintaining high availability and disaster recovery.

Skills – Golang, Spring Boot (2.x/3.x), Spring MVC, Spring Security, Spring Data JPA, Spring AOP, Spring DAO, Spring Context, Microservices Architecture, RESTful APIs, GraphQL, Apache Kafka (3.x), RabbitMQ (3.x), AWS Lambda, AWS SQS, AWS SNS, AWS EC2, AWS S3, AWS RDS, AWS DynamoDB, AWS API Gateway, AWS CloudFormation, Docker, Kubernetes (1.27+), CI/CD (Jenkins, GitHub Actions, GitLab CI/CD), PostgreSQL (14.x/15.x), MySQL (8.x), MongoDB (5.x/6.x), Redis (7.x), OAuth2, JWT (JSON Web Tokens), Spring Security, AWS IAM, AWS CloudWatch, Prometheus, Grafana, Terraform, AWS CDK, JUnit (5.x), Mockito (4.x), Cypress (12.x), Jest (29.x), React.js (17/18), Redux, Next.js, Angular (12/16), HTML5, CSS3, SCSS, Tailwind CSS (3.x), Material UI.

Full Stack Developer, Busy Human – Orem, Utah, United States Jun 2020 – Sep 2021

- Developed and maintained highly scalable **Java 11** backend services using **Django, Spring Boot 2.2**, **Spring MVC**, **Spring Security**, and **Spring Data JPA**, ensuring that business logic was efficiently processed while keeping response times low under high traffic conditions.
- Designed and implemented **RESTful APIs** and **GraphQL endpoints** that provided structured, flexible, and efficient data access to frontend applications, reducing latency and improving system reliability across microservices.
- Built and optimized responsive **React 16.13** front-end components using **Redux** for state management, **Material UI** for a polished user experience, and **TypeScript** for improved code maintainability, ensuring that sellers could efficiently track and manage their listings in real time.
- Led a major migration effort from **AngularJS to React 16**, redesigning core seller dashboard components to improve frontend performance, enhance maintainability, and reduce unnecessary re-renders by leveraging **React Hooks** and **functional components**.
- Implemented secure authentication and authorization mechanisms using **OAuth2**, **JWT**, and **Spring Security**, ensuring that seller accounts and sensitive data were fully protected while integrating with Amazon's enterprise identity management systems.
- Architected an **event-driven messaging system** using **Apache Kafka 2.4**, **AWS SQS**, and **SNS**, enabling real-time inventory updates and asynchronous order processing, which significantly improved system reliability and reduced backend load during peak sales periods.
- Designed and deployed **cloud-native microservices** on **AWS EC2**, **S3**, **RDS (PostgreSQL 12)**, and **DynamoDB**, integrating **AWS Lambda** for serverless event handling and reducing operational overhead by automating background tasks.
- Containerized microservices using **Docker**, deployed workloads in **Kubernetes 1.16**, and leveraged **Helm charts** to standardize deployments, ensuring a highly available and fault-tolerant architecture that could scale seamlessly during high-traffic events like **Prime Day**.
- Established robust **CI/CD pipelines** using **Jenkins** and **GitHub Actions**, automating the build, testing, and deployment process, which reduced release cycle times and allowed for quick rollbacks in case of production issues.
- Wrote extensive **unit tests** using **JUnit 5** and **Mockito 3**, integrated **Cypress 4.9** for end-to-end UI testing, and conducted **performance testing** with **JMeter**, ensuring that the application remained stable under varying loads.
- Led the **refactoring of legacy monolithic services** into a distributed **microservices architecture**, improving scalability, maintainability, and deployment flexibility while enabling faster time-to-market for new features.
- Designed and implemented **real-time monitoring and logging solutions** using **Amazon CloudWatch**, **Prometheus**, and **Grafana**, enabling proactive issue resolution and ensuring system uptime during high-traffic periods.
- Actively mentored junior developers and new hires, providing **technical guidance, conducting code reviews, leading architectural discussions, and organizing knowledge-sharing sessions**, which helped improve team efficiency and code quality.
- Worked closely with **product managers, UX designers, data engineers, and DevOps teams** to align technical implementations with business objectives, ensuring that new seller features were built with high performance, scalability, and user experience in mind.

Skills – Java 11, React 16.13, Spring Boot 2.2, Spring MVC, Spring Security, Spring Data JPA, Spring AOP, RESTful APIs, GraphQL, Apache Kafka 2.4, RabbitMQ, AWS Lambda, AWS SQS, AWS SNS, AWS EC2, AWS S3, AWS RDS (PostgreSQL 12), DynamoDB, API Gateway, CloudFormation, Docker, Kubernetes 1.16, Helm, CI/CD (Jenkins, GitHub Actions), Redux, TypeScript, Material UI, OAuth2, JWT, AWS IAM, CloudWatch, Prometheus, Grafana, JUnit 5, Mockito 3, Cypress 4.9, Jest, Postman, JMeter, Terraform, AWS CDK, HTML5, CSS3, SCSS, Tailwind CSS 3.x.

Software Engineer, Vention – New York, United States May 2016 – May 2020

- Developed and maintained backend services for **Amazon Fulfillment Web Services (AFWS)** using **Java 8, Python 3.6, Spring Boot 1.5, and Flask**, ensuring efficient order fulfillment and real-time inventory tracking for Fulfillment by Amazon (FBA) sellers.
- Designed and implemented **RESTful APIs** and **GraphQL endpoints** that facilitated seamless data exchange between warehouse management systems, third-party logistics providers, and Amazon's internal fulfillment network.
- Integrated **Apache Kafka 1.1** and **AWS SQS/SNS** for real-time event-driven communication, optimizing warehouse automation by reducing processing delays in inventory updates and order tracking.
- Built microservices architecture leveraging **Spring Boot, Flask, and AWS Lambda**, ensuring fault tolerance and scalability for handling high-volume order processing workloads.
- Developed predictive analytics tools using **Python (Pandas, NumPy, Scikit-learn)** to analyze shipment trends, optimize delivery routes, and enhance warehouse efficiency using machine learning algorithms.
- Deployed containerized services using **Docker** and managed deployments on **Kubernetes 1.11**, improving system resilience and enabling automated scaling for fulfillment applications.
- Implemented **authentication and authorization mechanisms** using **Spring Security, OAuth2, and AWS IAM**, ensuring secure access to fulfillment APIs and warehouse management tools.
- Conducted **unit and integration testing** using **JUnit 4, Mockito 2, PyTest, and Cypress 3.4**, ensuring backend stability and frontend reliability before production deployments.
- Developed real-time monitoring dashboards using **Amazon CloudWatch, Prometheus, and Grafana**, allowing operations teams to track system performance, identify bottlenecks, and improve response times.
- Mentored junior engineers by conducting **code reviews, architecture discussions, and best practices workshops**, fostering a culture of technical excellence and improving overall team efficiency.

Skills – Java 8, Python 3.6, Spring Boot 1.5, Spring MVC, Spring Security, Flask, RESTful APIs, GraphQL, Apache Kafka 1.1, AWS SQS, AWS SNS, AWS Lambda, AWS EC2, AWS S3, AWS RDS (PostgreSQL 10), DynamoDB, API Gateway, CloudFormation, Docker, Kubernetes 1.11, Helm, CI/CD (Jenkins, AWS CodePipeline), OAuth2, AWS IAM, JUnit 4, Mockito 2, PyTest, Cypress 3.4, Pandas, NumPy, Scikit-learn, Prometheus, Grafana, Amazon CloudWatch

Junior Microsoft Access Database Developer, CanNor – Ottawa, ON May 2012 – May 2016

- Developed and maintained **Microsoft Access databases** for tracking government funding applications and projects, ensuring accurate data entry, reporting, and retrieval for internal teams.
- Designed and implemented custom **queries, forms, and reports** to automate data processing, improving efficiency in managing program-related information.
- Collaborated with senior developers to **optimize database performance**, troubleshoot issues, and ensure data integrity across multiple departments.
- Assisted in migrating data from legacy systems to **Microsoft Access** and improved data accessibility for non-technical users through user-friendly interfaces.
- Provided **technical support and training** to end users, helping them navigate the database and generate necessary reports for internal audits and analysis.

Skills – Microsoft Access, VBA, SQL, Data Entry, Data Reporting, Query Design, Form Creation, Report Generation, Database Optimization, Troubleshooting, User Support and Training, Data Migration, MS Excel, Data Analysis.