

Rajkumar Kumaravelu

Boston, MA | (351) 220-5890 | kumaravelu.r@northeastern.edu

<https://rajkumarkumaravelu.netlify.app/> | <https://www.linkedin.com/in/rjvelu/> | <https://github.com/rjvelu>

Education

Northeastern University, Boston, MA - **Master of Science in Information Systems**

Expected May '24

Courses: Data Structures and Algorithms, Cloud Computing, Big Data & Distributed Systems, Web development

Technical Skills

Programming Languages: Java, JavaScript, Python

Web Technologies: HTML, CSS, Bootstrap, React.js, Redux, Node.js, jQuery, REST, Microservices, GraphQL

Database & Distributed Systems: SQL, NoSQL, MongoDB, Postgres, Redis, Elastic Search, Kafka

Cloud & CI/CD: AWS(S3, SNS, SES, DynamoDB, Route53), Pulumi, Packer, Jenkins, Docker, Kubernetes, GitHub Actions

Others: Git, Postman, Mockito, JUnit, OAuth, Bash, Agile & Scrum, JIRA, JMeter, Visual Studio, Prometheus, Grafana

Professional Experience

LeadHawk, Boston, USA

Jun' 23 – Dec' 23

Software Developer

- Implemented a **distributed system** using a **three-node Kafka cluster** on **AWS EC2**, distributed across multiple Availability Zones, to improve communication between services in a microservices architecture
- Created a **two-node Elasticsearch** cluster on AWS EC2 to facilitate real-time alerting, enhancing the efficiency of filter-based data retrieval and distribution
- Automated** the user filter matching for real-time alert dispatch on incoming data by using **percolate mapping**
- Dockerized** the app for scalability, subsequently deploying on AWS ECS for optimized service orchestration

National Payments Corporation of India, Chennai, India

Jan '22 – Aug '22

Associate Certification Operations – Software Development

- Programmed a scalable back-end login microservice with **Spring Boot** for certification tool, and utilized **Eureka Discovery Server** for interservice communication achieving twice lower latency between service communication calls
- Leveraged **Spring security** and implemented **OAuth2** and **JWT** tokens for secure authentication and authorization
- Designed the service using the **Builder pattern** to enhance the object construction needed for authentication
- Implemented **cache** feature using **Redis** for storing **public certificates**, reducing the retrieval times by 70% and significantly enhancing data validation efficiency
- Integrated **Hibernate** with Spring Boot using **Spring Data JPA** for Data Persistence to the Database and reduced the lines of code by 10%
- Conducted **load tests** using JMeter and created unit test cases using JUnit and Mockito

Straive, Puducherry, India

May '16 – Dec '21

Senior Subject Matter Expert – Software Development

- Utilized **Pub/Sub** model and created a message processing pipeline using **AWS SQS** that subscribed to a topic
- Established monitoring and real-time email alerts for dead letter queues using **serverless lambda functions** significantly minimizing downtime and eliminating backlogs by two times
- Analyzed** various databases to align with business needs, **evaluating retrieval speeds** and functionality for optimal service integration for the comments feature of an analytics app
- Developed a **responsive analytics dashboard** using React and implemented high-performance charts consuming **Chart.js** library capable of displaying 9000 data points
- Created reusable components using **React Hooks** and **Redux** for Central State Management, ensuring state consistency and increased efficiency by 10%
- Dockerized** a React and Spring Boot full-stack application, managed the lifecycle, and **deployed to Docker Hub**
- Built an In-app messaging **JAVA** application based on **STOMP protocol** which allows users to exchange instant messages on the Straive Platform
- Streamlined the reporting system using **iText library** and increased the number of reports generated by 5 times

Projects

Immutable Cloud Servers | Northeastern University

Sep '23-Dec '23

- Engineered assignment submission portal enabling a real-time notification system with AWS **RDS** for data storage
- Built customized AMI of Linux-Debian using **Packer** and deployed the application on **AWS EC2** instance
- Implemented **Elastic Load Balancer** to distribute traffic across Instances and **Auto Scalars** to scale instances
- Performed Continuous Integration using **GitHub Actions** and Continuous deployment using **Pulumi** with **zero downtime update** and **CloudWatch** for log collection and application monitoring