**Sai Krishna Mohan Kolla**

**SR. JAVA FULL STACK DEVELOPER**

**saikrishnakolla93@gmail.com | +1 251-721-7855 | [LinkedIn](https://www.linkedin.com/in/sai-krishna-kolla/) | [Portfolio](https://full-stack-skm.vercel.app/)**

**PROFESSIONAL SUMMARY:**

**Senior Java Full Stack Developer** with 10+ years of experience architecting and delivering enterprise-grade applications across banking, retail, fintech, and technology domains. Highly skilled in building scalable, secure, and distributed systems using **Java 17+, Spring Boot 3, Spring Cloud, Microservices, Kafka, and Node.js (Express)**, integrating REST and GraphQL APIs, and deploying cloud-native solutions on **AWS (EKS, ECS, Lambda, S3, RDS)** with additional experience on GCP and Azure. Adept at **frontend development** using **React 18, Redux Toolkit, Angular 8–15, TypeScript, JavaScript (ES6+), HTML5, CSS3, Bootstrap, and Material UI**, delivering responsive, accessible, and high-performance UIs. Strong in **database architecture and tuning** across **PostgreSQL, Oracle, MySQL, MongoDB**, and **Redis**, with exposure to **Cassandra** for distributed data management. Experienced in **DevOps and CI/CD pipelines** leveraging **Jenkins, GitHub Actions, Docker, Kubernetes, and Terraform** for automated build, test, and deployment workflows. Proven expertise in **modernizing legacy monolithic applications (Struts, JSF, EJB)** into **Spring Boot microservices** and React/Angular SPAs, improving scalability and maintainability by over 40%. Proficient in **test automation and quality engineering** using **JUnit, Mockito, Cucumber BDD, Selenium, Jest, and Cypress**, ensuring robust and SLA-compliant systems. Recognized for **technical leadership, architecture reviews, mentorship, and cross-functional collaboration**, consistently delivering secure, resilient, and cloud-optimized enterprise solutions that align technology excellence with business outcomes.

**TECHNICAL SKILLS:**

**Programming Languages & Frameworks:** Java 8 – 17, JavaScript (ES6+), TypeScript, Node.js, Express.js, SQL, Python (scripting)

**Backend & Microservices:** Spring Boot 3, Spring Core, Spring Cloud (Eureka, Feign, Gateway, Hystrix), Spring MVC, Spring Data JPA/Hibernate, Spring Batch, RESTful APIs, GraphQL (Apollo Server/Client), Microservices Architecture, Reactive Programming (WebFlux), Kafka Streams, RabbitMQ, JMS, WebSocket

**Frontend & UI Development:** React 18 (Hooks, Context API, Redux Toolkit, React Router), Angular 8–15 (RxJS, Angular Material, CLI), HTML5, CSS3, SASS/LESS, Bootstrap 5, Material UI, Ant Design, Tailwind CSS, Styled Components, jQuery, Chart.js, Recharts, D3.js, Highcharts, Formik + Yup, React Query

**Databases & Persistence:** PostgreSQL, Oracle, MySQL, MongoDB, DynamoDB, Redis (caching), Amazon Aurora, Cassandra (exposure), JDBC, Liquibase/Flyway (migrations), SQL tuning & indexing

**Cloud & Deployment:** Amazon Web Services (AWS EKS, ECS, Lambda, EC2, S3, RDS, CloudFront, CloudWatch, API Gateway, CodePipeline, CodeBuild, CodeDeploy), Google Cloud Platform (App Engine, Cloud SQL, Cloud Storage), Microsoft Azure (App Service, Azure Storage, Azure Data Factory, Azure DevOps Pipelines)

**DevOps, CI/CD & Automation:** Docker, Kubernetes, Terraform, Ansible, Jenkins, GitHub Actions, GitLab CI/CD, Maven, Gradle, Nexus, SonarQube, Shell Scripting

**Testing & Quality Engineering:** JUnit 4/5, Mockito, Cucumber (BDD), TestNG, Selenium, Jest, React Testing Library (RTL), Cypress, Postman, Swagger /OpenAPI, SonarQube, Allure Reports

**Monitoring & Observability:** ELK Stack (Elasticsearch, Logstash, Kibana), Splunk, Grafana, Prometheus, AWS CloudWatch, OpenTelemetry (exposure), Application Insights

**Version Control & Collaboration:** Git, GitHub, Bitbucket, JIRA, Confluence, Agile /Scrum Methodologies, TDD / SDLC Best Practices

WORK EXPERIENCE:

**Client : Fifth Third Bank | Dec 2023- present | Charlotte, NC**

**Sr. Java Full Stack Developer**

* Applied Solid principles and domain-driven design to re-architect legacy core-banking modules into Spring Boot 3 microservices, improving code modularity and transaction throughput by 35%.
* Developed loan origination and customer 360 microservices using Spring Cloud (Eureka, Gateway, Feign), Kafka Streams, and AWS EKS, enabling real-time data synchronization across channels.
* Implemented RESTful and GraphQL APIs to expose customer, loan, and payment data to multiple internal and partner applications, reducing integration time by 40%.
* Introduced a Node.js + Express.js Backend-for-Frontend (BFF) layer for React dashboards, optimizing API aggregation and cutting latency by 25%.
* Designed Spring Batch jobs for bulk loan reconciliation and settlement processing, achieving a 30% reduction in nightly batch duration and better error recovery.
* Implemented centralized authentication and authorization with Spring Security (OAuth2/JWT), enhancing security and reducing unauthorized access incidents by 20%.
* Deployed containerized microservices on AWS EKS/ECS with auto-scaling, load balancing (ELB), and Terraform automation, ensuring 99.9% uptime and predictable infrastructure costs.
* Automated CI/CD pipelines using Jenkins and GitHub Actions, integrated SonarQube and unit-test gates to enforce quality standards across 15+ services.
* Leveraged AWS Lambda and S3 for serverless file processing and document archival workflows, reducing manual storage tasks by 50%.
* Optimized data persistence with Hibernate and advanced indexing in Amazon Aurora and Redis, improving query performance and cache hit rates by 30%.
* Built interactive React 18 UI modules with Redux Toolkit and React Router, supporting role-based dashboards for customer service and loan officers.
* Adopted micro-frontend architecture to parallelize UI delivery across teams and enable independent deployment of React apps per business domain.
* Implemented real-time notifications via WebSocket channels backed by Kafka topics for payment status and customer alerts.
* Standardized logging and observability using ELK Stack (Elasticsearch, Logstash, Kibana) and AWS CloudWatch for centralized trace analytics and alerting.
* Conducted load testing and API benchmarking with JMeter and Gatling to validate performance under peak load, supporting >1 M daily transactions.
* Defined branching strategy (feature/hotfix/release) and pull-request governance in Git, reducing merge conflicts and release rollbacks by 40%.
* Collaborated with DevOps and cloud architects to adopt service mesh (Istio on EKS) for traffic management, policy enforcement, and zero-trust communication.
* Partnered with business and QA teams to convert UAT scenarios into Cucumber BDD scripts, increasing automated test coverage and reducing manual effort by 30%.
* Mentored junior developers in Spring Boot 3, React 18, and AWS best practices, strengthening team capability and delivery velocity.
* Acted as Scrum lead for multiple sprints, driving cross-team collaboration and ensuring timely delivery of critical core-banking enhancements.

**Tech Stack:** Java 17, Spring Boot 3, Spring Cloud (Eureka, Gateway, Feign), Spring Batch, Node.js, Express.js, React 18, Redux Toolkit, AWS (EKS, ECS, Lambda, S3, RDS, CloudFront, CloudWatch), Kafka, Redis, Aurora, Hibernate, Jenkins, GitHub Actions, Terraform, Docker, Kubernetes, ELK Stack, JMeter, Swagger, JUnit 5, Mockito, Cucumber, GitHub Actions, Agile/Scrum Methodology.

**Client: Huntington Bank | April 2021 –Nov 2023 | Columbus, OH**

**Sr. Java Full Stack Developer**

* Designed and implemented a **real-time payments and fraud detection platform** leveraging Spring Boot 2.7 microservices, Kafka Streams, and AWS ECS, enabling instant transaction validation and fraud event correlation across distributed systems.
* Architected REST and GraphQL APIs using Spring Boot and Apollo Server, providing unified access for payment gateways, audit teams, and external compliance services, reducing redundant integrations by 40%.
* Built a **rule-based fraud scoring engine** using Spring Batch and Java multithreading, processing over 1 million daily payment events with 99.98% accuracy and sub-second response times.
* Integrated **Kafka-based message pipelines** for transaction event streaming, improving system scalability and enabling predictive risk analytics.
* Developed and optimized PostgreSQL queries and schema-level indexing strategies, achieving a 35% improvement in query performance for high-volume transaction tables.
* Implemented OAuth2 and JWT-based authentication using Spring Security, ensuring secure multi-role access control for internal applications and APIs.
* Built React 17 web modules with Redux Toolkit for payment reconciliation dashboards, ensuring real-time updates and data consistency across audit and monitoring UIs.
* Containerized applications using Docker and deployed through AWS ECS with CloudFormation templates, ensuring consistency across development, QA, and production environments.
* Integrated **AWS Lambda** for asynchronous data validation and alert triggers, improving system responsiveness and reducing compute overhead.
* Configured AWS RDS and S3 for transactional storage, encrypted backups, and archival of processed payment data in compliance with PCI DSS.
* Automated CI/CD pipelines using Jenkins and Gradle, embedding static code analysis and automated testing gates to enforce continuous quality.
* Adopted ELK Stack and AWS CloudWatch for centralized logging, monitoring, and proactive alerting on payment anomalies and application performance.
* Enhanced fraud reporting efficiency by building scheduled batch jobs to aggregate high-risk transactions and push daily audit summaries to compliance systems.
* Collaborated with data science teams to embed ML-based anomaly detection models into microservices for real-time fraud identification.
* Conducted performance and load testing using JMeter and optimized thread pools, improving throughput and reducing CPU utilization by 20%.
* Documented APIs using Swagger/OpenAPI and standardized REST conventions across 20+ microservices, reducing onboarding time for new developers by 30%.
* Implemented BDD test automation using Cucumber integrated with Jenkins, ensuring end-to-end test coverage and consistent release quality.
* Led daily Agile stand-ups and sprint retrospectives, aligning cross-functional deliverables between development, QA, and DevOps teams.
* Mentored junior engineers on microservice best practices, React state management, and AWS cloud deployments, improving team efficiency and velocity.
* Partnered with enterprise architects to modernize core payments platform from monolithic WebLogic applications to containerized Spring Boot microservices hosted on AWS ECS.

**Tech Stack:** Java 11, Spring Boot 2.7, Spring Batch, Spring Security (OAuth2/JWT), React 17, Redux Toolkit, GraphQL (Apollo Server/Client), Kafka, PostgreSQL, Redis, AWS (ECS, Lambda, RDS, S3, CloudWatch, CloudFormation), Jenkins, Gradle, Docker, ELK Stack, JMeter, Swagger, JUnit 5, Mockito, Cucumber, Git, Agile, Scrum, JIRA.

**Client : Walmart | July 2018- March 2021 | Dallas, TX**

**Lead Java Full Stack Developer**

* Led the design and implementation of a **global retail inventory and order management system** using Spring Boot microservices and Angular 8, supporting 50K+ concurrent transactions across North American and APAC regions.
* Architected and developed RESTful APIs with Spring Boot 2.x, Spring Data JPA, and PostgreSQL, enabling real-time inventory synchronization and reducing order latency by 40%.
* Designed microservice communication patterns leveraging **RabbitMQ message queues** and Kafka topics to ensure fault tolerance and consistent event delivery between inventory, pricing, and fulfillment domains.
* Migrated legacy Oracle WebLogic and monolithic JSP applications into a **containerized microservices ecosystem** using Docker and Kubernetes (GKE), improving scalability and deployment efficiency by 45%.
* Built rich, interactive dashboards and admin portals in Angular 8 with RxJS, Angular Material, and dynamic data visualization components (Chart.js, Recharts), enhancing data transparency and warehouse decision-making.
* Integrated **Redis caching** for real-time product availability lookups, achieving sub-100 ms response times for high-traffic product pages.
* Implemented service discovery and configuration management using Spring Cloud, improving service resiliency and centralized configuration across 30+ microservices.
* Automated CI/CD pipelines using Jenkins and Maven, integrating SonarQube for static code analysis and JUnit tests to ensure 90%+ test coverage on core modules.
* Deployed microservices to **Google Cloud Platform (GCP)** using App Engine, Cloud SQL, and Cloud Storage, optimizing cost and improving scalability with autoscaling policies.
* Collaborated with security teams to integrate OAuth2-based authentication and authorization, protecting APIs and user sessions with enterprise SSO.
* Implemented API Gateway and rate-limiting strategies to handle peak seasonal loads during global sales events, ensuring uninterrupted service uptime.
* Designed and implemented distributed caching and database sharding for large-scale SKU data, reducing replication lag and query load on master databases.
* Conducted extensive performance tuning of Spring Boot and Angular applications, cutting memory overhead by 25% and improving user time-to-first-byte metrics.
* Standardized logging and observability through ELK Stack (Elasticsearch, Logstash, Kibana) integrated with GCP’s Stackdriver for centralized log analysis.
* Coordinated Agile sprints across backend and frontend teams, managing delivery of 8 developers while mentoring junior engineers in microservice design, Git branching, and Angular architecture.
* Documented system architecture, API contracts, and deployment runbooks for internal DevOps and support teams, improving operational readiness.
* Collaborated with data analytics and BI teams to integrate order and inventory events into downstream reporting pipelines for sales forecasting and supply-chain optimization.
* Championed the migration of batch ETL processes into Spring Batch workflows, reducing nightly processing times by 30%.
* Delivered multiple high-visibility features, including real-time inventory tracking and global order visibility dashboards, ahead of schedule and within performance SLAs.

**Tech Stack:** Java 8, Spring Boot 2.x, Spring Cloud, Spring Data JPA, Angular 8, RxJS, Material UI, PostgreSQL, Redis, RabbitMQ, Docker, Kubernetes (GKE), Google Cloud Platform (App Engine, Cloud SQL, Cloud Storage), Jenkins, Maven, SonarQube, ELK Stack, OAuth2, JUnit, Mockito, Git, Bitbucket, and Scaled Agile (SAFe).

**Client : Amazon | may 2017 – June 2018 | Seattle, WA**

**Java Full Stack Developer**

* Developed and maintained **warehouse automation and fulfillment modules** to manage order processing, shipment routing, and inventory tracking across multiple regional hubs, improving operational throughput by 25%.
* Engineered RESTful APIs and event-driven services using **Spring Boot 1.x, Spring MVC, and Hibernate**, enabling seamless integration between inventory, logistics, and order management systems.
* Designed and deployed **asynchronous workflows** using AWS **SQS and SNS** to handle order confirmations, tracking updates, and warehouse notifications, reducing message delivery latency by 35%.
* Implemented **AWS Lambda functions** for event triggers and background job processing, improving scalability and cutting manual job scheduling overhead by 40%.
* Integrated React JS 16 dashboards for warehouse and logistics operations, providing real-time visibility into shipment statuses and process metrics.
* Utilized AWS **S3 for document and image storage**, enforcing encryption-at-rest and lifecycle management for cost optimization.
* Enhanced API response times by applying **connection pooling, Hibernate second-level caching, and query optimization**, improving overall system performance under high transaction volumes.
* Refactored legacy SOAP-based web services into RESTful endpoints, improving maintainability and simplifying integration with partner systems.
* Implemented user authentication and authorization flows via **Spring Security** and JWT tokens to secure internal APIs and React frontends.
* Conducted unit and integration testing using **JUnit and Mockito**, ensuring code reliability and preventing regressions across builds.
* Automated builds and deployments with **Jenkins**, integrating with AWS CodePipeline for continuous delivery and blue-green deployment strategies.
* Partnered with DevOps teams to containerize microservices using Docker, ensuring consistent deployment environments across test and production.
* Implemented **AWS CloudWatch and ELK Stack** for application monitoring, error tracking, and proactive issue resolution, reducing production incidents by 20%.
* Collaborated with product managers and cross-functional teams to design scalable REST endpoints and UI components aligned with business KPIs.
* Conducted API load and regression testing with Postman and JMeter, validating system stability under peak holiday traffic.
* Documented architecture, API endpoints, and operational procedures, improving knowledge transfer and onboarding efficiency for new developers.
* Participated in Agile sprint planning, reviews, and retrospectives, ensuring sprint deliverables aligned with performance and scalability objectives.
* Contributed to modernization roadmap by introducing reusable microservice templates and modular UI libraries, accelerating feature delivery for downstream teams.

**Tech Stack:** Java 8, Spring Boot 1.x, Spring MVC, Hibernate, React JS 16, AWS (EC2, S3, Lambda, SQS, SNS, RDS, CloudWatch, CodePipeline), Oracle DB, Maven, Docker, Jenkins, JUnit, Mockito, ELK Stack, Postman, JMeter, Git, Kanban workflow, and CI/CD automation culture.

**Client :upstox | June 2015 – July 2016 | Mumbai, IN**

**Jr. Java Software Engineer**

* Developed and maintained complex Model-View-Controller (MVC) architectures using the Struts framework, enabling separation of concerns and efficient management of application layers.
* Integrated AWS services (EC2, Lambda, RDS, S3) to build scalable, serverless solutions, reducing infrastructure costs by 25%, while designing and deploying cloud-based database and file storage systems that improved data retrieval speed by 30% and ensured secure, reliable data management.
* Utilized SQL query optimization techniques and managed database performance to ensure fast and efficient data retrieval, contributing to overall system performance improvements.
* Migrated Struts 1/JSP applications to Spring Boot, redesigning backend services and modernizing legacy workflows for better scalability and maintainability.
* Designed and implemented business logic components using Enterprise JavaBeans (EJB) to manage transactional operations and ensure scalability, while collaborating with senior developers and business analysts to translate requirements into reliable, client-focused solutions.
* Developed and consumed SOAP-based web services to integrate with external systems, ensuring seamless platform communication and data exchange, improving integration efficiency by 25%, and reducing data transfer errors.
* Participated in SDLC activities, including designing, implementing, and testing Java-based applications, ensuring adherence to coding standards and best practices.
* Adopted Git for version control, standardizing workflows and reducing merge conflicts by 30%.
* Conducted thorough code reviews to maintain high-quality standards, identify potential issues, and ensure compliance with industry best practices.
* Optimized HTTP request/response handling to improve web service performance and user experience, achieving a 20% reduction in response time, while implementing monitoring and logging frameworks to ensure proactive issue detection and faster resolution.
* Applied expertise in transaction management to ensure data integrity and consistency in multi-user environments, leveraging EJB and other Java technologies.
* Implemented robust error handling and logging mechanisms using Log4j, improving application stability and simplifying troubleshooting and monitoring tasks.
* Built proof-of-concept modules in Struts and JSP that were later scaled into production features.
* Documented technical workflows and knowledge transfer for junior teammates, improving onboarding speed by 15%.

**Tech Stack**: Struts, AWS EC2, AWS Lambda, AWS RDS, AWS S3, SQL, JavaServer Pages (JSP), Java Servlets, Enterprise JavaBeans (EJB), SOAP, Git, Log4j.

CERTIFICATIONS:

* MERN Stack Developer - Digital Lync
* ChatGPT Prompt Engineering for developers - DeepLearning.AI