|  |  |
| --- | --- |
| **Rachana Chowdary**  rachanareddychowdary@gmail.com  **+1 (989) 294-48355**  **Sr. Full Stack Java Developer** |  |

### **Professional Summary:**

### Experienced Sr. Full Stack Java Developer with over 10 years of expertise in **AWS Cloud-Native Application Development**, specializing in backend services using Java, Spring Boot, and microservices. Strong experience in **AWS infrastructure provisioning (CloudFormation, Terraform), Kubernetes (EKS/ECS), and cloud database technologies (RDS, DynamoDB)**. Skilled in secure API design and **authentication protocols including SAML, OAuth 2.0, and JWT**. Proficient in DevOps and **CI/CD** automation using **Jenkins**, **GitHub**, and AWS pipelines. Hands-on experience with **Python** for automation and AWS Lambda scripting. Excellent problem-solving, detail-oriented, and effective in **Agile** environments with strong communication and leadership skills.

### **Technical Skills:**

* **Programming Languages:** Java, JavaScript, TypeScript, Python
* **Frameworks/Libraries:** Spring, Spring Boot, Spring MVC, Spring Data JPA, Spring Cloud, Spring Security, Spring Batch, Hibernate, Microservices, React, Redux, Angular
* **Web Technologies:** HTML5, CSS3, Bootstrap, AJAX, jQuery
* **Databases:** Oracle, MongoDB, MySQL, DynamoDB, RDS, SQL, NoSQL
* **Cloud Platforms:** AWS, Azure
* **Tools:** Git, GitHub, JIRA, Docker, Kubernetes, Terraform, Jenkins, Maven, Gradle
* **Other Technologies:** Kafka, RabbitMQ, GraphQL, CI/CD, RESTful Web Services
* **Testing:** JUnit, Mockito, Cucumber, Jest, Mocha
* **Methodologies:** Agile (Scrum, Kanban), Test Driven Development (TDD)

**Work Experience:**

**Sr. Full Stack Java Developer** **03/2024 - Current**

**GEICO**, **Remote**

* Developed **Java** based microservices using **Spring Boot** to process real-time insurance policy updates, customer onboarding, and claims workflows, ensuring high system throughput and fault tolerance.
* Built and optimized **RESTful services** to integrate **billing**, **underwriting**, and customer portals, reducing data sync latency.
* Implemented **Kafka**-based event streams to process live claim events and trigger downstream actions like fraud detection and audit logging.
* Built middleware services using **Java** to parse insurance documents and generate structured data for downstream processing.
* Utilized **Java** concurrency features to handle asynchronous customer data processing efficiently.
* Implemented global exception handling and validation logic in **Spring Boot** controllers for better error reporting.
* Designed and optimized **MongoDB** **schemas** using Java-based MongoDB drivers to improve query performance and reduce response times.
* Implemented data aggregation and complex queries using **MongoDB** for real-time analytics.
* Integrated **MongoDB Atlas on AWS** with Spring Boot microservices to support real-time analytics and policy data processing.
* **Designed and developed highly interactive user interfaces using React, utilizing React Hooks and Context API to optimize state management and improve user experience.**
* Implemented state management with **Redux** in React applications, improving application performance and maintaining scalable data flow.
* Integrated **RESTful APIs** developed in **Java** into **React** frontend, ensuring seamless data retrieval and synchronization with backend services.
* Designed and optimized schemas in **Amazon DynamoDB** for customer onboarding workflows, ensuring low-latency lookups and scalable performance.
* Integrated **Amazon S3** for scalable and cost-effective storage solutions, ensuring efficient data accessibility for Java-based applications.
* Integrated **AWS Lambda** functions written in **Java** with **S3**, **DynamoDB**, and **API Gateway** for seamless data processing and real-time event handling.
* Deployed **Java**-based microservices using **AWS ECS** and **EKS**, ensuring scalability and high availability for containerized applications.
* Configured **AWS** **SNS** to send notifications for critical application events, enhancing real-time monitoring and alerting.
* Implemented **OAuth 2.0** authentication and authorization mechanisms using **Java** and **Spring Security** to enhance security and compliance with industry standards.
* Integrated **SAML**-based SSO for enterprise identity management alongside OAuth 2.0 and JWT authentication.
* Implemented **CI/CD** pipelines with **Jenkins** and **GitHub**, automating deployment processes and streamlining deployment efficiency.
* Provisioned **AWS** infrastructure using **CloudFormation** templates for **ECS/EKS** clusters and networking resources, ensuring scalable and secure deployments.
* Implemented **Kubernetes Ingress controllers** to manage efficient load balancing and routing across distributed services.
* Automated unit and integration testing with **JUnit** and **Mockito**, improving test coverage and code reliability.
* Conducted **code reviews** and implemented best practices for **Java** and **React** codebases, enhancing code quality and maintainability.
* Collaborated with cross-functional teams to ensure compliant and **robust solutions** for policy management and claims processing.

**Environment**: Java, Spring Boot, RESTful APIs, Microservices, JavaScript, React, Redux, MongoDB, AWS, Kafka, Docker, Kubernetes, Jenkins, Git, GitHub, Maven, JUnit, Mockito, OAuth, Agile

**Sr. Full Stack Java Developer** **01/2022 - 12/2023**

**Morgan Stanley**, **CA**

* Developed a high-performance **Java/J2EE** application using **Spring Boot** and **Spring MVC**, enhancing real-time **trade processing and customer portfolio management.**
* Developed a batch processing system using **Spring Batch** with **Java** to handle large volumes of financial transactions with improved efficiency.
* Integrated third-party financial APIs seamlessly into **Java-based Spring Boot applications**, ensuring secure and reliable transaction processing.
* Implemented **MongoDB** transactions with **Spring Data MongoDB** using Java, ensuring data consistency and integrity across complex banking transactions.
* Integrated **Kafka** producers and consumers to handle real-time transactional data feeds, ensuring low-latency updates for banking operations.
* **Created reusable React components** that interact with **RESTful APIs**, ensuring modularity and consistency across multiple banking applications.
* Optimized frontend performance by leveraging lazy loading and code splitting techniques in **React** applications, utilizing **HTML**, **CSS**, and **JavaScript** for responsive interfaces.
* Integrated **OAuth** **2.0** and **JWT** for token-based authentication with **Spring Security** inJava, providing secure API access and user authentication for RESTful APIs.
* Implemented **AWS RDS PostgreSQL with multi-AZ failover** for high-availability financial transaction systems.
* **Integrated multi-factor authentication (MFA)** in internal and customer-facing applications to enhance security.
* Implemented **SAML 2.0 authentication** in Spring Security to enable secure identity federation with banking partners.
* Configured backend services to consume **SAML assertions and metadata** for role-based access control in trading applications.
* Configured **AWS SNS** and **SQS** for trade notification services, ensuring timely client updates and operational efficiency.
* Utilized **AWS Lambda** with Java for serverless computing, optimizing cost-efficiency and scalability for **event-driven microservices.**
* Managed automated database backups and disaster recovery strategies using **AWS** **S3** and **RDS** snapshots, for critical Java-based financial systems.
* Orchestrated **microservices** deployments with **Docker** and **Kubernetes**, enhancing scalability and reliability of banking application.
* Orchestrated Spring Boot microservices on **Kubernetes**, leveraging **Spring Cloud Gateway** for efficient routing in a financial services architecture.
* Automated **AWS** infrastructure provisioning using **Terraform** for **Lambda**, **RDS**, and **SQS** resources to support financial transaction workloads.
* Implemented monitoring and logging for **EKS** **clusters** to ensure smooth operation and quick resolution of issues.
* **Automated deployment pipelines** with **AWS CodePipeline** and **Jenkins**, enhancing CI/CD workflows and improving deployment efficiency.
* Developed automated unit tests with **JUnit** and **Mockito**, ensuring the robustness of critical banking features through a **TDD** approach.
* Actively engaged in **Agile** ceremonies, including sprint planning, backlog refinement, and retrospectives to drive team-wide continuous improvement.

**Environment**: Java/J2EE, Spring Boot, Spring MVC, Spring Batch, Spring Security, RESTful APIs, Microservices, JavaScript, React, Redux, MongoDB, AWS, Kafka, GraphQL, Docker, Kubernetes, Jenkins, JUnit, Mockito, TDD

**Full Stack Developer 09/2018 - 12/2021**

**PayPal**, **IND**

* Developed high-performance backend services using **Java** and **Spring Boot** for processing large volumes of payment transactions and enabling real-time analytics.
* Implemented and optimized RESTful APIs using **Spring MVC** to support seamless integration with frontend platforms and third-party payment services.
* Designed microservices architecture with **Spring Boot** and **Spring Cloud** to modularize payment and risk management systems for better scalability and maintainability.
* Applied in-memory caching using **Java Map** and **Concurrent HashMap** to improve response times and reduce repeated database calls.
* Integrated **MongoDB** with **Spring Data MongoDB** to store and manage transaction data, utilizing schema validation and custom query methods.
* Improved **MongoDB** performance through effective indexing strategies, enhancing query speed in high-traffic payment systems.
* Built responsive web interfaces using **React**, enabling real-time updates and smooth user experience during transactions.
* Developed reusable navigation components with **React Router** for consistent routing and user flows across the payment platform.
* Utilized **CompletableFuture** and **Spring @Async** for handling real-time transaction updates with non-blocking operations.
* Deployed **Spring Boot** microservices on **AWS** using **EC2**, **Lambda**, **S3**, and **RDS** to achieve scalability and fault tolerance.
* Tuned **SQL** queries in **PostgreSQL** using **Spring Data JPA** and **Hibernate** to ensure fast and reliable data access.
* Performed unit and integration testing with **JUnit 5** and **Mockito** to validate functionality and ensure code reliability.
* Configured Java microservices to validate **SAML tokens** for secure communication with third-party payment providers.
* Built scalable RESTful APIs with **Spring Boot** and exposed them via **AWS API Gateway** for low-latency payment request handling.
* Configured **AWS CloudWatch** for real-time monitoring of payment systems, enabling proactive issue detection and resolution.
* Containerized **Spring Boot** applications using **Docker** and deployed them with **Kubernetes (EKS)** for automated orchestration and scaling.
* Provisioned AWS RDS, S3, and API Gateway resources using **CloudFormation templates**, reducing manual setup and ensuring environment consistency.
* Automated deployment of containerized Spring Boot services with **Terraform modules**, streamlining Kubernetes (EKS) provisioning.
* Automated **CI/CD** pipelines with **Jenkins**, **Maven**, and **GitHub Actions** to automate build, test, and deployment workflows.
* Implemented real-time data streaming with **Apache Kafka** using **Spring Kafka** and monitored with **Prometheus** and **Grafana** for reliability.

**Environment**: Java, Spring Boot, Spring MVC, Spring Data JPA, Spring Cloud, REST APIs, MongoDB, PostgreSQL, React, Redux, JavaScript, AWS, Docker, Kubernetes, JUnit, Mockito, Jenkins, Maven, Microservices, Kafka, SQL

**Java Developer** **07/2015 - 09/2018**

**Oracle,** **Hyderabad, India**

* Developed scalable **Java** applications using **Spring** Framework, leveraging **dependency injection (DI) and inversion of control (IoC)** for modular and maintainable architecture.
* Implemented **RESTful** web services and APIs using **Spring MVC** for seamless communication between distributed systems.
* Built and deployed **cloud-native microservices** using **Spring Boot**, integrating with **Oracle Cloud** for seamless scalability and high availability.
* Utilized **Java 8 Streams API** and **Lambda Expressions** for efficient data processing, functional programming, and robust error handling.
* Secured applications with **Spring Security in Java**, implementing **authentication, authorization, and role-based access controls.**
* Configured and secured **Apache Tomcat**, implementing SSL certificates and access control mechanisms to protect sensitive data.
* Utilized **Hibernate ORM and Spring Data JPA** for efficient database access, optimizing **SQL** **queries** and schema design for **Oracle** databases.
* Implemented **Oracle PL/SQL** stored procedures and triggers to enhance data processing efficiency and enforce business rules at the database level.
* Integrated **Redis** based caching mechanisms, to reduce database load and improving response times.
* Implemented **DAO design pattern** to separate data access logic, ensuring modularity and improved database interaction.
* Implemented **multithreading** and **concurrency** solutions in Java to enhance application performance.
* Leveraged **AWS Lambda** and **API Gateway** to build serverless RESTful APIs, ensuring scalability and cost efficiency.
* Utilized **Amazon RDS** and **DynamoDB** for optimized database performance and seamless integration with Java applications.
* Deployed microservices on **AWS EKS**, implementing CI/CD pipelines with **AWS CodePipeline** for automated deployments.
* Built dynamic front-end applications with **Angular**, **HTML**, **CSS**, and **TypeScript**, delivering responsive user interfaces.
* Integrated **Angular** components with **Spring REST** controllers and consumed external APIs using HttpClient for real-time data integration and third-party service interactions.
* Ensured code quality and reliability of Java applications by following **Test-Driven Development (TDD)** practices with **JUnit and Mockito**.
* **Implemented CI/CD pipelines** using **Oracle Cloud DevOps** services for automated build, test, and deployment processes.
* Utilized **Python** scripting for automating build and deployment tasks, enhancing pipeline efficiency with **Maven**.

**Environment**: Java 8, Spring, Spring MVC, Spring Data JPA, Spring Boot, Spring Security, RESTful APIs, GraphQL, Hibernate, Oracle, PL/SQL, Angular, Typescript, Apache Tomcat, Maven, JUnit, Mockito, **Oracle Cloud**, AWS, Design Patterns.

**Education:**

**Geethanjali University** Hyderabad, India.

**Bachelor of Technology** in Computer Science and Engineering 2015