**RISHWANTH**

**P: (475)289-6059 || E: rishwanthreddy282@gmail.com**

 **PROFESSIONAL SUMMARY**

* **Fullstack Java Developer** with around **10 years** of experience in analysis, design, implementation, and testing of Object-Oriented Applications using **Core Java, Servlets, JDBC, JSP, Hibernate, Spring,** and **Web Services** with a proven track record of integrating technical processes within high-profile, and tight deadline-oriented environments.
* Expert in Various methodologies like **SCRUM, Test Driven Development (TTD)**, Incremental and Iteration methodology, **Agile Development** & **Testing** using **Software Development Life Cycle**.
* Experienced in the Development, Testing, and Deployment of enterprise applications on **Windows, Linux,** and **UNIX** platforms using **IDEs** such as **Eclipse, NetBeans,** and **IntelliJ 14.x/15. x.**
* Well versed with **Design patterns** - **Data Access Object, Session Façade, Business Delegate, Service Locator, Data Transfer Object, Singleton**. Also with **Java** features - **Lambda expressions, functional interfaces, Stream API, Date & time APIs, ZGC (Z Garbage Collector).**
* Proficient in developing **UI** using **JavaScript, TypeScript, jQuery, AJAX, Angular (JS, 8, 9, 13, 16, 17), Node JS, React JS (React Context API, React Hooks, Functional Components,** middleware - **Redux Thunk, Redux-Saga).**
* Created high-performance, reusable UI components using **Angular 16, NgRx,** and **TailwindCSS,** integrating with **REST** and **GraphQL endpoints** via **Apollo Client** and **HTTP interceptors.**
* Worked on designing role-based **UI/UX flows** integrating **Keycloak/Okta** and secured with **OAuth2 SSO, refresh tokens,** and access policy introspection.
* Expert in Architecturing and deploying scalable, secure **microservices** using **Spring Boot, Kafka, REST/GraphQL APIs, Redis,** and **PostgreSQL** across multi-region **Kubernetes clusters** on **AWS, Azure, GCP, PCF** environments**.**
* Experience with **AJAX frameworks** such as **DOJO,** and **JSON** in designing integration components using **JMS** and **MQ**.
* Proficient in **Spring Framework** modules such as **Spring MVC, IOC, AOP, JDBC, JTA, IO, ORM, Spring Dashboard, Spring Boot, Spring Microservices, Spring REST, Spring Eureka, Spring Ribbon, Spring Zuul Proxy.**
* Experience in d**esign architecture** and **development** of batch jobs using **Spring** **Quartz** and **Spring batch.**
* Led backend system modernization from **monolith to distributed microservices** using **DDD, CQRS,** and event-drivenarchitecture with **Kafka, Spring Cloud Stream,** and **Confluent Schema Registry.**
* Built **microservices** using **Spring Cloud, Kafka, REST,** and **GraphQL,** following **DDD, CQRS, 12-Factor principles.**
* Developed enterprise applications using **EJB, Webservices (SOAP, REST, JAX WS, JAX RS, Apache Axis, WSDL, UDDI), Rule Engines (Drools), JBPM tools, Jersey.** Proficient in developing secure and scalable **APIs** using **Spring Security, OAuth2, JWT,** and **TLS/mTLS,** deployedbehind **Apigee Edge** and **Spring Cloud Gateway.**
* Proficient in Developing and consuming **RESTful** and **GraphQL APIs,** versioned with **OpenAPI 3.0** and secured using **Spring Security, OAuth2.0, JWT,** and **mTLS** across internal and external boundaries.
* Well versed with **Middleware** persistence frameworks like **Hibernate/JPA** Entities for **mapping Java classes** using **Hibernate Query Language (HQL), HSQL Named Queries, Criteria,** and **Projections.**
* Architected **event-driven** systems with **Kafka, Kafka Streams, Schema Registry,** and asynchronous messaging via **Spring Cloud Stream** and **RabbitMQ, ActiveMQ, Zookeeper.**
* Implemented **observability** and **telemetry** using **Prometheus, Grafana, OpenTelemetry, Spring Boot Actuator, and** centralized structured logging with **Fluentd/ELK.**
* Well versed with **Camunda BPM** as Delivered **BPM-enabled workflows** using **Camunda BPM,** embedding decision tables **(DMN) a**nd orchestrating user tasks with custom **Angular** interfaces**.**
* Expert in writing testable, maintainable code with **JUnit 5, Mockito, TestContainers, REST Assured, Cypress,** and **Pact** for E2E testing. Also, performed unit testing using **Selenium, Cucumber, Junit, Mockito, Mock MVC, JMeter, J-Test**.
* Tuned **JVM** applications using **JFR, async-profiler, G1GC,** and **runtime metrics profiling** for optimal throughput.
* Enforced **API standards** and design consistency across teams with shared **DTO models, OpenAPI,** and **API governance.**
* Experience in working on **Maven, Gradle,** and **Ant** for project build and logging tools such as **Log4J2** and **Splunk**
* Strong experience with **MongoDB** development **(reports, schema design, map-reduce functions)** and migrations from **SQL** relational databases to **NoSQL** databases like **MongoDB, Cassandra,** and **CouchDB.**
* Advanced SQL expertise with **PostgreSQL 15**, **MySQL 8**, **SQL Server 2022**, and NoSQL databases including **MongoDB**, **DynamoDB**, **Cosmos DB**, **Redis**, and **Cassandra** for distributed data modeling and high availability.
* Expert in **configuring** and **deploying** applications into **Tomcat, WebSphere, WebLogic, JBoss,** and **Glassfish Servers.**
* Worked with **Docker** in creating **Docker images**, **Containers, push/pull** and **integrating** with **Spring Boot**. Used **Kubernetes** on multiple clouds to build and containerize their application **(CI/CD) pipelines** for deployment.
* Built fully automated **CI/CD** pipelines using **GitHub Actions, Jenkins,** and **Terraform,** integrating **SonarQube, Trivy,** and artifact repositories for secure, stable releases.
* Containerized services with **Docker** and orchestrated deployments on **Kubernetes (GKE/EKS/AKS)** using **Helm**, **Kustomize**, and **GitOps** tools like **ArgoCD** and **Flux**.
* Extensive experience designing and deploying scalable, secure, and event-driven architectures on **AWS** using services like **EC2, S3, RDS, Lambda, API Gateway, IAM, SQS, Kinesis, VPC** and infrastructure as code with **Terraform, AWS CDK, CI/CD** using **CodeBuild/CodeDeploy,** and centralized logging with **ELK** stack forhigh-availability microservices**.**
* Proficient in architecting and deploying cloud-native applications on **Microsoft Azure**, leveraging **Azure App Services, AKS, Azure Functions, Cosmos DB, Azure SQL, Blob Storage, Key Vault** and worked on **Azure DevOps pipelines**, **ARM/Bicep templates**, and secure networking via **VNet, NSG, Application Gateway** for automated deployments.

 **TECHNICAL SKILL**

|  |  |
| --- | --- |
| **Languages** | Java 1.8/8/11/15/17, C, C++, SQL, PL/SQL, UML |
| **Java Technologies** | Microservices, EJB, Servlets, JSP, JSTL, JAX-RS, JDBC, JNDI, JMS, JAXB, JUnit, Mockito, AWT, Spring MVC, Spring AOP, Spring DAO, Spring Core, Spring Security, Spring Actuator, Spring Boot, JPA, Hibernate, Spring JDBC, Spring Data JPA, Spring Data REST, Log4j, ActiveMQ, RabbitMQ, Apache Kafka, Netflix OSS, JMeter. |
| **Web Technologies** | HTML5, CSS3, Bootstrap, JSP, AJAX, XML, JSON, Node JS, Express JS, ReactJS, Angular JS, 9/11/13/16 and Karma, Jasmine, Chai, and Jest |
| **Web/Application Servers** | Tomcat, WebLogic, IBM WebSphere, JBOSS |
| **Design Patterns** | MVC, Front Controller, Session Façade, Singleton, and DAO patterns |
| **DataBase** | Oracle, MS SQL Server, MySQL, DB2, NoSQL - MongoDB, Cassandra |
| **Web Services** | SOA architecture, RESTFUL/SOAP, Microservices, Axis, Apache CXF, GraphQL, Swagger |
| **Build & Other Tools** | Rational Rose, CVS, Git, SVN, VSS, Clear Case, JIRA, Jenkins, JUnit, TestNG, Selenium, Cucumber, Jbehive, Mockito, IBM MQ, Log4j, ANT, Maven, Docker, Kubernetes, Gradle, Camunda 7/8.7 (BPMN, DMN, Zeebe, Modeler). |
| **Cloud Technologies** | AWS EC2, S3, RDS, AWS Lambda, AWS Cognito, SQS & SNS, AWS Identity and access management, CloudWatch & EKS, Azure (SQL, ADD, Logic Apps, APIM, Cosmos DB, Blob Storage, VMs, Service Bus, Azure Monitor, & ARM). |

 **PROFESSIONAL EXPERIENCE**

**Verizon, NYC, NY Dec’ 23 – Till date**

**Sr. Java Full Stack Developer**

**Responsibilities:**

* Developed Microservices-based applications using Spring, Spring Boot, Hibernate, Spring DAO, Gradle, JDBC, Oracle.
* Integrated Angular 16 services with RESTful APIs, Angular 14 HTTP Client to send GET/POST requests to fetch data from the server. Configured Microservices architecture and integrates with React JS application.
* Leveraged Angular 16 for data access, created custom directives, factories, services for components and Used Angular 16 components like modules, controllers, synchronized models with single-page Application.
* Developed components in Angular 16 and customized from Angular libraries to meet the application's functionality.
* Created reusable and testable React JS functional and class components using ES6.
* Performed Reverse Engineering to map POJO classes to database. Used Eureka to call other proxies Microservices.
* Integrated new APIs for list processing introduced in Java 17, including List.copyOf() and List.of() methods, for creating immutable lists with concise syntax and improved performance.
* Worked on protocols (such as WSDL and SOAP), Service Oriented Architecture (SOA), development of XSD schemas, and implementing both SOAP and REST web services.
* Developed high-throughput RESTful APIs using Spring Boot, MVC, resource models, URI naming, response contracts.
* Designed and implemented distributed backend systems using Spring Boot microservices, each owning its own domain model and database, following bounded context and domain-driven design (DDD) principles.
* Developed RESTful APIs with Spring MVC, applying HATEOAS principles via URI and header-based strategies.
* Engineered custom exception handling middleware using Spring’s @ControllerAdvice and response entity wrappers, standardizing error formats across microservices.
* Implemented asynchronous service communication using Spring Events and RabbitMQ/Kafka, enabling decoupled interactions between internal microservices.
* Used Hibernate Entity Graphs, fetch joins, second-level caching with Ehcache to improve JPA & avoid N+1 query issues.
* Migrated monolithic modules into REST microservices for APIs using REST Template/WebClient for inter-service calls.
* Created annotations and AOP-based interceptors for cross-cutting concerns like logging, validation, and authorization.
* Tuned JDBC, JPA performance by setting batch, fetch sizes, and analyzing query plans with Oracle, MySQL EXPLAIN.
* Exposed microservices through Google Apigee API Gateway, applying custom API proxies, quota policies, spike arrest, and message transformation rules to control external traffic.
* Versioned APIs via header and URI strategies for backward compatibility using Swagger, OpenAPI 3.0 for integration.
* Implemented rate-limiting, caching, JWT verification policies in APIGEE compliance with partner SLAs, reducing load.
* Enabled GraphQL support for composite queries in select services for UI performance by minimizing API round-trips.
* Applied database migration strategies using Flyway, Liquibase for consistent schema evolution across environments.
* Configured and maintained Helm charts and OpenShift YAML manifests to automate deployment of stateless and stateful Java applications across dev, staging, and production clusters.
* Used Spring Retry, Resilience4j to build fault-tolerant microservices with retries, circuit breakers, fallback mechanisms.
* Optimized performance using custom thread pools, connection pooling (HikariCP), lazy-loading strategies in Spring Boot.
* Integrated Prometheus and Grafana with OpenShift monitoring stack to visualize JVM metrics (heap, GC, thread usage) from Java services via Micrometer and Spring Actuator.
* Participated in capacity planning and performance tuning for Java 17 services running on OpenShift using JVM tuning flags and OpenShift resource limits/requests.
* Wrote step definitions to develop BDD tests using Cucumber. Developed Selenium support code in JAVA for Cucumber.
* Used JSP as a view layer with MVC to integrate with Servlets, JavaBeans to separate business logic with presentation.
* Implemented asynchronous programming with Node JS (Promises, Async/Await) for improved non-blocking operations.
* Used Maven for project dependency management and Log4j for logging Involved in activities like code review, performance improvement and used Jira for bug reporting.
* Worked with unit testing of Angular applications using Jasmine and Karma and Spring Boot applications using JUnit, Mockito and performed end to end testing using Protractor.
* Wrote unit and integration tests using JUnit to perform integration testing manually checking the flow between microservices and Mockito, employing parameterized tests, spies, argument captors, and test coverage analysis.
Built test automation into CI pipelines using SonarQube for code quality metrics, test thresholds to gate deployments.
* Managed containerized services using Docker & deployed to Kubernetes clusters with Helm for zero-downtime rollouts.
* Integrated Kafka message pipelines with MongoDB, Oracle for polyglot persistence, using outbox and CDC patterns.
* Applied Kafka partitions and consumer groups to parallelize processing and improve throughput, tuning offsets and poll intervals for maximum efficiency.
* Conducted REST API contract testing and schema validation using Postman, REST Assured, and JSON Schema Validator.
* Architected and deployed microservices on Azure Kubernetes Service (AKS) and Azure App Services, integrating with Azure SQL, Cosmos DB, and Blob Storage for transactional and unstructured data handling across global regions.
* Designed event-driven serverless solutions using Azure Functions with triggers from Azure Event Grid, Azure Blob Storage, and Cosmos DB Change Feed, supporting asynchronous workflows, file ingestion, and real-time notifications.
* Built secure, scalable APIs using Azure API Management (APIM) with OAuth2 enforcement, throttling policies, and backend routing to AKS and App Services; secured sensitive data using Azure Key Vault and managed identities.
* Automated infrastructure provisioning with ARM templates and Bicep, deploying VNets, subnets, NSGs, App Gateways, AKS clusters, and PaaS services using parameterized, reusable IaC modules.
* Implemented full CI/CD pipelines in Azure DevOps, integrating unit testing, SonarQube, security scans, approvals, and blue/green deployments, ensuring reliable and auditable multi-environment delivery.
* Developed a new build system using Jenkins, uDeploy, and Maven for code deployment and to improve build quality.

**Environment:** Java 17, Angular 16, Hibernate, Spring Boot, Spring MVC, Microservices, JavaScript, JQuery, Bootstrap, Ajax, SOAP/REST, Jenkins, PCF, GIT, Kubernetes, Docker, Mockito, JUnit, Jasmine, Karma, Apache CXF, Oracle, MySQL, NoSQL, MongoDB, PL/SQL, Maven, SonarQube, Azure (AKS, API, CosmosDB, SQL, AppServices, Blobstorage, ARM, VNets, App Gateways, Key Vault), Log4j, Kubernetes, Agile.

**US Bank, Minneapolis, MN Jan’ 22 – Nov ‘23**

**Sr. Java Full Stack Developer**

**Responsibilities:**

* Optimized Java 15 garbage collection using G1GC and ZGC for improved application performance under heavy load.
* Enhanced performance with Foreign Function & Memory API (FFM) for efficient interaction with native libraries.
* Involved in Construction of UI using HTML5, CSS3, Ajax, TypeScript, ES6 (ECMAScript), JQuery, Angularjs, Ext JS, ReactJS, Redux, Bootstrap and JavaScript and worked on server-side web applications using Node.js.
* Worked in using React JS components, Forms, Events, Keys, Router, Animations and Flux concept. Used React-Router to turn application into a Single Page Application.
* Worked on React Profiler, Chrome DevTools to resolve performance bottlenecks in component rendering, state updates.
* Used Hibernate and MarkLogic, writing service methods and SQL queries to interact with the MySQL DBMS.
* Worked with NPM commands and using package.json for managing dependencies, and dec-dependencies in Node JS.
* Involved in ORM mappings for associations, Inheritance & fetching strategy, performing CRUD operations and managing second level cache using Hibernate.
* Used JavaScript with asynchronous programming techniques, including Promises and async/await, to optimize API calls.
* Integrated API calls using Axios and implemented parallel API calls to reduce transition time in the application.
* Implemented modules using Core Java APIs, Java Collections, Streams, multi-threading, object-oriented designs.
* Optimized API Gateway configurations to enable high availability, load balancing, automatic failover and Integrated API Gateway with Lambda for executing serverless functions and reducing backend server load.
* Exposed RESTful and GraphQL APIs through API Gateway, ensuring better data efficiency and flexibility.
* Secured Mainframe-Java hybrid environments using TLS, OAuth2, Kerberos, and LDAP-based authentication.
* Automated API security audits using SonarQube, Fortify, identifying and mitigating security vulnerabilities proactively
* Developed Microservices using REST API with Java, Spring Boot, Express and used Maven as a build tool.
* Implemented functionality using Spring Boot and Hibernate ORM to map with Oracle Database.
* Used Spring components like Spring MVC, Spring AOP, Spring Batch, Spring IOC, Spring Transactions and Spring Security modules for developing the service layer of the web application.
* Used Spring Boot - Microservice interacting through a combination of REST and Apache Kafka message brokers.
* Used Apache Kafka Queues for reliable and asynchronous exchange of information between multiple services.
* Implemented fail safe and circuit breaker patterns in Spring Boot Microservice applications using Hystrix and monitored the services using Hystrix Dashboard.
* Involved in implementing the Microservices based on RESTful API utilizing Spring Boot with Spring MVC.
* Used Spring-boot-starter-parent as the parent POM and inheriting sensible default settings from Spring Boot and the spring-data- Mongodb dependency provides integration with the MongoDB.
* Developed Java connectors using Kafka, WebSphere for asynchronous message processing with Mainframe applications.
* Used Spring Configuration files to create the beans needed and injected dependency using Dependency Injection.
* Implemented thread-safe data structures including ConcurrentHashMap, CopyOnWriteArrayList, and BlockingQueue to facilitate concurrent access and modification in multi-threaded environments.
* Implemented API key validation, JWT verification, and OAuth token validation using Apigee Edge security policies to enforce API security policies and prevent unauthorized access to APIs.
* Used Spring Boot for cloud Microservices and to develop Spring application radically faster with very less configuration.
* Used Apache CXF to develop SOAP, RESTful services, Swagger API for auto-generated documentation for REST calls.
* Used SOA Integration service for Data Integration Service, Functional Integration Service and used SOAPUI for the functional testing, inspection and invocation of Spring based Web services.
* Integrated Apache Flink with Kafka for fault-tolerant streaming analytics, enhancing data synchronization and ensuring consistent processing across distributed systems.
* **Implemented Business Process Management (BPM)** logic for loan origination and customer onboarding using Camunda’s task listeners, process variables, and incident handling features to maintain system consistency.
* Designed BPMN workflows using **Camunda 8.7**, modeling complex business logic including multi-instance tasks, timers, gateways, and compensation events to streamline approval and escalation processes.
* Created integrations with Camunda REST APIs for workflow orchestration, communication between internal tools & supported process monitoring and Microservices to communicate using protocols HTTP, REST for implementing SOAP.
* Implemented Oracle SQL Queries & PL/SQL - stored procedures, functions, triggers, sequences and built-in functions to retrieve and update data from the databases.
* Implemented Apache Kafka producer & consumer for asynchronous stream processing, search APIs using data indexing.
* Deployed and tested the application in UNIX on IBM Web Logic Application Server.
* Managed Docker orchestration using Docker Swarm and containerization of applications using Docker.
* Monitored MongoDB (NoSQL) Replica Sets / Databases and prototyped MongoDB new features across multiple stacks.
* Working HA Environment, Configuring the Kubernetes pods and testing the pods.
* Enabled Continuous Integration, Continuous Deployment (CI/CD) and monitoring using Chef, Jenkins and Shell Scripts.
* Involving the code review and using the SonarQube to validate the violation.
* Configured Jenkins and GitHub Actions for CI/CD pipelines, automating build, test, and deployment workflows.
* Used Jenkins as Integration tool to deploy Spring Boot Microservices to Pivotal Cloud Foundry (PCF) using build pack.
* Integrated testing methods implemented by using Karma and Jasmine. Used Jira as bug reporting tool.
* Deployed serverless applications using AWS Lambda, API Gateway, and DynamoDB, reducing operational overhead.
* Automated infrastructure provisioning and management using AWS CloudFormation, Terraform, enabling consistency.
* Configured AWS KMS for managing encryption keys across AWS services for secure data storage and compliance with GDPR and other data protection regulations.
* Implemented CI/CD pipelines using AWS Code Pipeline, Code Build, and Code Deploy for testing and deployment.
* Automated backup processes for critical AWS resources (EC2, RDS, S3) using AWS Backup and Lambda, ensuring data availability and compliance with retention policies.
* Deployed workflow-driven microservices to **AWS Elastic Kubernetes Service (EKS)** using Helm, configuring autoscaling policies, IRSA (IAM Roles for Service Accounts), and service mesh (App Mesh) for secure and scalable orchestration.
* Optimized performance by leveraging AWS services like Elastic Cache, CloudFront, RDS Read Replicas.
* Developed unit test cases using JUNIT, Easy Mock, Integration and User Acceptance Testing and developed test cases for bean components. Used Maven, Jenkins to build and deploy applications.

**Environment:** Java 15, MVC, Spring Boot, React JS, Redux, Flux, Node JS, Backbone JS, Hibernate ORM, MongoDB (NoSQL), PCF, REST/SOAP Web services, JUNIT, Microservices, Web Logic, Maven, JBOSS, JavaScript, BPM, Camunda 8.7, AWS (EC2, EKS, S3, RDS, Lambda, DynamoDB, API Gateway), SQL-Server, Log4j, Jenkins.

**Liberty Mutual, Boston, MA Sep’ 19 – Dec’ 21**

**Java Full Stack Developer**

**Responsibilities:**

* Developed RESTful APIs using Spring Boot with robust integrations for banking applications, optimizing scalability and performance. Spring Boot's microservices architecture facilitated seamless deployment and maintenance.
* Implemented stable React components and stand-alone functions. Used React for Building component libraries like Tree, Slide-View, Table Grid using React strap, React-Router, Axios service to retrieve data from server synchronously.
* Developed Microservices (REST) using Spring Boot, Spring Data with JPA (Java Persistence API), deployed Microservices on Docker. Worked on REST API, invoked web services (consumer) based on EJBs.
* Integrated Angular 11 services with RESTful APIs, Angular 11 HTTP Client to send GET/POST requests to fetch data from the server. Configured Microservices architecture and integrates with Web applications.
* Created TypeScript reusable components and services to consume REST API using Angular 11 component architecture.
* Implemented MVC design pattern to organize Angular 11 controllers, custom directives, factories, and views.
* Designed robust workflows with Spring Data JPA, improving query precision and minimizing transaction delays. Spring Data JPA's repository features accelerated development cycles.
* Integrated Swagger using Springfox and Spring MVC, enhancing API documentation usability. Springfox allowed for automated documentation generation and interactive testing.
* Enhanced authentication mechanisms using OAuth2, integrating token-based authorization for seamless third-party API access and leveraging the extensibility of OAuth2 for enterprise-level security.
* Implement a JPA repository to access data as DAO. Implement Kafka for both Producer and Consumer.
* Tested REST APIs with GET, POST, PUT, and DEL by providing input in JSON parameters using the Postman.
* Implemented API Gateway to manage, secure, and monitor RESTful and GraphQL APIs in a microservices architecture.
* Implemented request/response transformation and traffic throttling to enhance API security and performance.
* Defined infrastructure as code with Terraform, ensuring consistent environment setups across development, staging, and production. Terraform accelerated provisioning and deployment times.
* Managed API lifecycle by creating, deploying, versioning, and retiring APIs using AWS API Gateway and APIGEE.
* Developed custom Wavemaker widgets, modules, integrating into Angular applications to enhance functionality.
* Deployed business workflows using Camunda, enabling seamless orchestration of complex processes and improving system scalability with BPMN, DMN, and CMMN standards.
* Integrated OpenShift with third-party tools and services (e.g., LDAP, SSO) to enhance authentication, authorization, and monitoring capabilities. Wrote JUnit test cases and used Mockito for mocking service calls.
* Deployed Spring Boot - Microservices Docker container using AWS EC2 container, AWS admin console.
* Used SoapUI, JMeter to check load balancing in Web Logic with multi-thread and to check accounts.
* Developed and maintained comprehensive unit tests, integration tests, and end-to-end tests for Angular applications using tools like Jasmine, Protractor or Cypress.
* Leveraged Mockito annotations such as @Mock and @InjectMocks to simplify test setups and improve test readability.
* Integrated Swagger using Springfox and Spring MVC, enhancing API documentation. Springfox allowed for automated documentation generation and interactive testing.
* Utilized Kubernetes tools and services such as kubeadm, kubectl, and Helm for efficient cluster management.
* Integrated SonarQube with Jenkins to track code coverage before moving deployment of the tag to production.
* Integrated OpenShift with CI/CD pipelines (Jenkins, GitLab CI) to streamline application builds, testing, and deployments and Managed OpenShift clusters using OpenShift CLI (OC) and Kubernetes APIs, performing tasks such as node scaling, resource allocation, and upgrade procedures.
* Used AWS for deployment on EC2 instances for MongoDB Services. Deployed Microservices as RESTful Java Web Services on AWS Elastic Beanstalk and AWS Connect for efficient connection with clients and setting up voice, chat, and tasks for contact routing, queuing, analytics, and management.

**Environment:**Java 11/15, Spring, MVC, Spring Boot, Spring Security, jQuery, Bootstrap, Hibernate, Angular 11/9, React, JavaScript, AJAX, JPA, JSF, JSP, OAuth2, Apache Kafka, AWS EC2, S3, AWS Lambda, Microservices, Servlet, RESTful, Docker, Kubernetes, SQL, Oracle 12c, MongoDB, GIT, JUnit, Maven, Log4j, JIRA, Jenkins, Swagger, Postman.

**Hexagon, Pune, India Oct’ 15 – Sep’ 18**

**JAVA/J2EE Developer**

**Responsibilities:**

* Developed multiple enterprise applications, which are developed using Spring MVC framework, Hibernate framework, Prime faces, RESTful web services.
* Designed and developed UI using spring view component HTML5, CSS3, Bootstrap, AngularJS and Vue JS.
* Injected Angular 4 built-in HTTP service to custom services to communicate with backend with RESTful APIs and Used Pipes to format and sort data displayed in the templates.
* Used Angular CLI to generate components, services, filters and compiled in different environments.
* Built custom UI widgets, directives, filters, and Application logic using Angular JS MVC Architecture.
* Implemented Hibernate Template to great extent making use of HQL, Named Parameters, Named Queries, and interfaces such as Expression, Query and Criteria.
* Used Multithreading and Concurrency and implemented threads while preparing and generating mails.
* Developed Hibernate with Spring Integration as data abstraction to interact with the database of MongoDB.
* Involved in developing applications using Angular JS like Grids, Toolbars, Panels, Combo-Box and Buttons.
* Used Hibernate, object/relational-mapping (ORM) solution and technique of mapping data representation from MVC model to Oracle Relational data model with SQL-based schema.
* Used Core Java concepts like Exception Handling, Collection APIs to implement features & enhancements.
* Created web pages using Angular data driven and template driven forms and created custom asynchronous validations to validate the data on server side. Developed UI Screen to display alerts in a grid using jQuery and Bootstrap.
* Created and consumed Restful/SOAP Web Services which was developed based on Spring MVC and Apache CXF API.
* Used Angular 2 as framework to create a Single Page Application (SPA) which can bind data to specific views and synchronize data with server. Wrote database triggers, procedures, functions in PL/SQL.
* Used Maven as build tool and deploying the application, Developed Jenkins job to deploy projects automatically in QA and Production environments.
* Set up Jenkins server, build jobs to provide continuous automated builds based on polling GIT during the day, scheduled builds overnight to support development needs using Jenkins, GIT, JUNIT, Selenium, Maven.
* Worked on Lambda Expressions, Functional interfaces Stream API's, Time API, and Improvements on Collection, Concurrency, and IO improvements by using Java 8.
* Developed test cases in JUnit/Mockito, Maven builds to achieve functionality in the build process.
* Used a Microservice architecture, with Spring Boot-based services interacting through REST, Apache Kafka.
* Developed Microservices using Spring Boot, Netflix OSS, Spring Cloud and deploying to AWS Cloud.
* Deployed Spring Boot based microservices Docker container using AWS EC2 container services.
* Created Stacks using AWS Cloud Formation templates to launch AWS Infrastructure and AWS Lambda to run the code.
* Used Orchestration container Kubernetes for load balancing and to schedule tasks across clusters and load balancers for backend API services. Implemented system logging solution using Elastic Search, Kibana and Logstash (ELK).
* Created RESTful services with Spring RESTful, sending JSON data between Frontend, middle tier controller
* Developed, Maintained server-side code with JavaScript and Node JS runtime environment.
* Performed deployment of applications on WebSphere Application Server Application Server.
* Deploying and maintaining a production environment using AWS EC2 instances, AWS EKS and ECS with Docker.
* Designed and developed a project that reads messages from Kafka topic and applied conversation from JSON to pipe delimited String data format to be saved in Oracle DB and NoSQL DB.
* Used Apache Maven for project management and building the application.
* Used Log4j utility to generate run-time logs. Used Clear Case for project and version management.
* Implemented microservices container CI/CD within AWS leveraging Jenkins, Git-hub, Docker, Ansible, Kubernetes.
* Used Junit, Mockito frameworks to write the unit test and used the Emma plugin to measure code coverage.

**Environment:** Java, Spring, Hibernate, Spring batch, Spring security, Spring Boot, JPA, JavaScript, AngularJS, HTML5, CSS3, Node JS, NoSQL, MongoDB, log4j, Junit, Selenium, SOAP UI, Kafka, Kubernetes, AWS.

**Ides Labs, India May’ 13 – Sep’ 15**

**Node JS Developer**

**Responsibilities:**

* Designed and implemented high-performance microservices using Node JS (Nest JS, Express JS) with TypeScript.
* Implemented next-gen authorization by integrating Plaid, Stripe, PayPal, and Square APIs for secure transactions.
* Developed privacy-first applications with encryption, differential privacy, and zero-knowledge proofs in Node.js.
* Designed Angular forms and dynamic dashboards integrated with RxJS observables and Angular services to visualize real-time customer data.
* Implemented reusable UI components, validations with role-based field-level access control for regulatory compliance.
* Developed Java services for onboarding, KYC verification, and fund transfers, integrating UI and backend services.
* Implemented multi-threaded, concurrency processing in Node JS using Worker Threads, Web Workers, child processes.
* Built RESTful APIs using Spring Boot secured with JWT and OAuth2 for customer, account, and transaction management.
* Created custom policies in API Gateway for traffic shaping, IP whitelisting, and request validation to enhance security.
* Implemented request/response transformation policies to standardize API communication across backend services.
* Implemented OAuth 2.0 authentication & authorization using APIGEE OAuth policies to secure API endpoints and control access to sensitive resources, integrating with identity providers such as Okta, Auth0, LDAP.
* Integrated Node JS applications with Redis, Elasticsearch, MongoDB for caching, full-text search, NoSQL data storage.
* Designed progressive web applications (PWAs) with Node.js and React UI, optimizing performance and offline support.
* Developed Node.js APIs with rate limiting and circuit breakers using Rate-limiter-flexible to enhance fault tolerance.
* Integrated React-based biometric and identity verification modules into Angular using micro frontend architecture.
* Configured Kafka producers and consumers to publish transaction logs and customer updates in real-time.
* Deployed microservices architecture using Spring Cloud Gateway, Eureka Discovery, and Config Server for dynamic routing and resilience.
* Wrote and optimized SQL procedures and queries on Oracle DB for transaction summaries and KYC reports.
* Implemented CI/CD pipeline with Jenkins and GitHub Actions, including Docker-based builds and Sonar checks.
* Achieved 90%+ code coverage with JUnit and Mockito unit and integration test suites.
* Deployed services on AWS ECS with auto-scaling and configured CloudWatch for monitoring and alerting.
* Built DB servers using AWS EC2, Lambda, used Oracle DB for user-defined types, inheritance, and polymorphism.
* Implemented IAM policies, roles, permissions running on AWS, ensuring granular access control, security compliance.
* Implemented load balancing, auto-scaling using AWS Auto Scaling Groups, Kubernetes HPA, Nginx reverse proxies
* Utilized Log4j2 for centralized logging and applied static code analysis with SonarQube.
* Collaborated with product owners and QA in Agile sprints, performing daily standups, retrospectives, and demos.

**Environment:** Node.js (ES6+), TypeScript, JavaScript, Express.js, NestJS, Hapi.js, RESTful API, GraphQL, gRPC, OAuth2, WebSockets, OpenAPI (Swagger), PostgreSQL, Redis, Cassandra, Docker, Kubernetes, Jenkins, Terraform, Prometheus, HELM, ELK Stack, Splunk, DataDog, OpenTelemetry, TLS/SSL, OWASP, Jest, Mocha, Supertest, Selenium, GitHub, Jira, Confluence, Kafka, AWS SQS/SNS, AWS (Lambda, ECS, API Gateway, DynamoDB, S3)

 **EDUCATION**

* Master’s in Information Technology from Oklahama University, OK
* Bachelor’s in computer science from Andhra University.