**Varaha Krishna (682) 320 3766 |** varahakrishna94@gmail.com | [LinkedIn](https://www.linkedin.com/in/varaha-krishna/)

**SUMMARY**

* 6+ years of experience in designing, developing, and deploying scalable web and microservice-based applications using **Java 8/11/17**, **Spring Boot**, **REST APIs**, and **AWS**, with domain exposure in finance, healthcare, retail, and logistics.
* Proficient in **Spring** ecosystem, including **Spring MVC, Spring Security, Spring REST,** and **Spring AOP** for building on-prem and cloud-native enterprise applications.
* Strong background in **event-driven systems** using **Apache Kafka**, **Apache Flink**, and **JMS** for real-time data streaming, pub-sub architecture, and asynchronous communication.
* Experience in **containerization** and **container orchestration** using **Docker**, **Kubernetes**, and **OpenShift**.
* Experience building **DAG orchestration** workflows on Apache Airflow and Azure Data Factory.
* Built end-to-end **DevOps pipelines** using **GitLab**, **Jenkins**, and **SonarQube** to enable automated testing and high-quality CI/CD deployments for microservices.
* Skilled in core **Java** features like **streams**, **lambda** expressions, **annotations**, and functional interfaces.
* Developed large-scale persistence layers using **Hibernate**, **Spring JDBC/JPA**, and **DAO** pattern.
* Developed **RESTful APIs** using **Spring REST, GraphQL, OpenAPI,** and **Swagger**;secured using **OAuth2.0** and **PingFederate**.
* Strong experience applying **design patterns** like Singleton, Factory, Builder, Observer, Command, Decorator, and Proxy.
* Integrated Java applications/services with external systems such as **Maximo ERP, ServiceNow, and SalesForce,** applying **Enterprise Integration Patterns**.
* Demonstrated leadership in driving backend and full-stack development teams, frequently mentored developers, and led design and code reviews.
* Enforced high code quality, modular service architecture, and reusability through **design reviews,** **SonarQube quality gates,** **JUnit** test cases, and up-to-date service/API documentation.
* Implemented centralized logging and monitoring using the **ELK Stack** (Elasticsearch, Logstash, Kibana) and observability tools like **Grafana** and **Prometheus** for real-time diagnostics.
* Expert database design, modeling, PL/SQL, and optimization skills in **RDBMS** and **NoSQL databases**, including **Oracle, SQL Server,** **MySQL**, **Cassandra, and MongoDB**.
* Experienced in high-velocity **Agile team** delivery, driving **sprint cycles,** cross-functional team coordination, and dependency management in fast-paced environments.
* Performed **data preprocessing and transformation** using **Python (Pandas, NumPy)** to clean, normalize, and structure large datasets for reporting and analytics pipelines.
* Built **Tableau dashboards** and **Cognos reports** using SQL-based data, enabling KPI tracking and actionable insights for various business reporting teams.
* Collaborated with business analysts to convert reporting requirements into **dynamic BI dashboards**, enabling faster decision-making and reducing manual reporting efforts.
* Proven experience improving **speed-to-market** for high-complexity delivery initiatives.

**EDUCATION**

**MS** in Computer Science,University of Texas at Arlington

**B.Tech** in Computer Science, Jawaharlal Nehru Technological University, Hyderabad

**EXPERIENCE**

**Sr. Java Developer | Wells Fargo, New Jersey Aug 24 – Present**

The project aimed to modernize Wells Fargo’s retail digital banking platform by developing microservices to support real-time transaction processing, account activity tracking, and personalized customer notifications. The initiative replaced legacy systems with event-driven microservices using Spring Boot, Spring WebFlux, and Cassandra, enhancing responsiveness, scalability, and audit transparency. It enabled seamless integration with online and mobile banking channels, allowing customers to view and act on transactions instantly while ensuring compliance with internal risk and reporting policies.

**Responsibilities:**

* Modernized legacy transactional services into modular **Spring Boot microservices**, enabling real-time fund transfers, payment posting, and account updates with improved fault isolation and scalability.
* Built **async APIs** using **Spring Rest** and **Kafka messaging** to support high-throughput transaction processing and reduce customer-facing latency during peak loads.
* Developed **RESTful APIs** using Spring Boot’s annotation-based configuration and layered architecture, improving code maintainability.
* Optimized data persistence using **Hibernate** with domain-driven design principles, improving access to transactional and audit records used by internal reporting systems.
* Developed **PL/SQL-based reconciliation batches**, ensuring data integrity and accuracy across transaction lifecycles.
* Enabled faster release cycles and reduced deployment failures by automating critical service static code analysis, testing, and deployment using **JUnit**, **SonarQube**, **Maven**, and **Jenkins**.
* Built data import jobs to ingest large-scale datasets (>1 TB) from various sources to the Cassandra cluster using **Apache Flink** for data pipelines and **Jython/Python** for data transformations**.**
* Utilized **Spring Boot Actuator** for health checks, environment configuration, and custom metrics, improving monitoring and incident response times.
* Orchestrated containerized deployments using **Docker** to improve portability across development and staging environments, reducing onboarding time for new developers.
* Played a key role in improving system reliability, reducing downtime, and supporting millions of transactions daily with sub-second latency.
* Used the **Singleton pattern** to manage shared services like configuration and logging across microservices for consistent behavior and efficiency.

**Environment:** Java 17, Spring Boot, Spring Rest, Spring Security, Hibernate, RESTful APIs, Swagger, Oracle, PL/SQL, Cassandra, Docker, Jenkins, Maven, Git, JUnit 4/5, Mockito, Apache Flink, Jython/Python, Oauth2.0

**Backend Developer | St. Jude Children’s Hospital-ALSAC, Tennessee Jun 2023 - May 2024**

The project involves modernizing the donor and fundraising management system to securely handle online donations, automate donor engagement workflows, and provide real-time tracking of fundraising campaign performance. It will also ensure high availability, data security, and real-time metrics tracking.

**Responsibilities:**

* Migrated core donation and fund-processing applications from a **monolithic architecture to distributed microservices** using **Spring Boot,** significantly reducing release cycle times and enhancing scalability.
* Developed and exposed **RESTful APIs** using **Spring Rest** while integrating **OpenAPI/Swagger** for API documentation, ensuring external teams could onboard faster.
* Created **model transformers using Java 11 Stream API** to efficiently map backend DTOs, improving data serialization across service layers.
* Orchestrated donation services using **AWS EKS** and **HPA**, achieving 100% uptime during national campaigns.
* Integrated services with **AWS API Gateway** and **APIGEE** for throttling, security, and analytics, improving API governance and access control.
* Built **Kafka streaming pipelines** to track **real-time donation flows**, supporting transparent fundraising dashboards and alerts.
* Enabled **advanced donor search and segmentation** via Elasticsearch Query DSL, improving internal team efficiency by 50%. Enabled **advanced donor search and segmentation** via Elasticsearch Query DSL, improving internal team efficiency by 50%.
* Developed responsive frontend components using **React.js** and integrated them with Spring Boot APIs to enhance real-time donor engagement and streamline campaign tracking workflows.
* Worked with **AWS Glue DynamicFrames** for batch data transformation and loading to support reporting needs, ensuring accuracy in financial records.
* Configured **Prometheus and Grafana** for application-level observability; set up proactive alerts and dashboards to monitor JVM metrics, request latency, and error rates.
* Streamlined donation platform CI/CD using GitLab, reducing manual overhead by 70%
* Configured **Apache Kafka** consumer and producer metrics to visualize and monitor **Kafka** system performance.

**Environment:** Java 11, Spring Boot, Spring Security, Swagger/OpenAPI, Apache Kafka, Apache Flink, Elasticsearch, AWS (EKS, Lambda, Glue, API Gateway), Kubernetes, Helm, HPA, Python, Shell, Jenkins, GitLab CI/CD, Prometheus, Grafana, Mockito

 **Java Developer | Berkshire Hathaway Energy Dec 2021- Aug 2022**

The project aimed to modernize the utility billing system using microservices to calculate customer bills, track daily energy usage, and validate payments. It replaced a slow legacy system and improved performance. The system also made it easier for customer service teams to access real-time account and billing data.

**Responsibilities:**

* Engineered **Spring services** to decouple business functionalities such as billing calculation, usage analytics, and payment validation, improving deployment independence and resilience.
* Developed **RESTful** Web Services using Spring MVC, secured with **Spring Security** for role-based access and session management, ensuring restricted access to energy account data.
* Integrated **Spring Cache** to reduce latency in high-volume billing data retrieval by 40%.
* Used **Spring AOP** for centralized logging and monitoring of API interactions, helping teams trace and audit customer service workflows.
* Processed energy consumption and billing datasets using **Java 8 Streams and Lambdas** for efficient batch reconciliations.
* Created and managed **stored procedures** for business rule enforcement in Oracle, triggered from Hibernate to maintain consistency across database operations.
* Implemented **real-time search** on billing history data using Elasticsearch, enhancing customer support experience.
* Designed and developed **Tableau dashboards** to visualize energy consumption trends and billing KPIs, enabling business users to monitor performance and make data-driven decisions.
* Automated utility billing service releases via **Jenkins** pipelines, enabling weekly releases
* Developed **JUnit** and **integration test suites** to validate service layer and database interactions, boosting test coverage and ensuring high system reliability under load.
* Deployed Spring Boot microservices on **AWS EC2** with integrated **CloudWatch monitoring** and **S3 archival**, ensuring compliance with data retention policies while optimizing infrastructure cost and scalability.
* Supported **UAT and production deployment**, collaborating with QA and operations teams to resolve issues rapidly and ensure smooth system handoffs.
* Leveraged **JAX-RS and Spring REST** for RESTful interface design, supporting **JSON and XML**, to integrate with external payment processing and usage analytics vendors.

**Environment:** Java 8, Spring MVC, Spring Security, Hibernate, Spring AOP, Spring Cache, REST APIs, JAX-RS, Elasticsearch, Oracle, JDBC, AWS EC2, AWS S3, Jenkins, GitHub, JUnit, Lambda Expressions.

**Java Developer | FedEx Jun 2018 - Dec 2021**

The project focused on building a real-time logistics and shipment tracking system for FedEx using scalable services to handle shipment status updates and delivery event streaming. It replaced batch-based workflows with event-driven architecture using Kafka and Apache Flink. It enabled up-to-the-minute package visibility and improved last-mile delivery accuracy, system reliability, and operational efficiency.

**Responsibilities:**

* Developed and deployed real-time logistics and shipment tracking microservice modules using **Spring Boot** and **Spring REST**, improving last-mile delivery accuracy and package visibility.
* Utilized **Spring JDBC Templates** with **Oracle** to manage configuration and master data, optimizing access to large payloads related to shipment demand and supply.
* Architected and deployed a **three-node clustered MongoDB setup** with shard keys for distributed data storage, ensuring high availability and fault tolerance.
* Handled asynchronous communication via **AJAX** using **Spring RestTemplate API** and integrated **Spring JPA** for ORM-based relational data handling.
* Implemented centralized logging using **Logstash** and performed log analysis and defect fixing with **Kibana**, reducing issue resolution time by 25%.
* Followed **Test-Driven Development (TDD)** using **JUnit4**, **Mockito**, **H2**, **Embedded Mongo**, **MockMvc**, and **MockRestServiceServer** to ensure robust unit, DAO, and end-to-end tests.
* Built and scheduled **Cognos reports** by integrating SQL-based data sources, improving visibility into daily work orders.
* Streamlined CI/CD processes using **Maven** for dependency management and **Jenkins pipelines** for automated builds, testing, and deployments, cutting release cycles by 30%.
* Implemented the **factory** **pattern** to instantiate shipment handlers dynamically based on delivery type, streamlining routing logic across logistics microservices.
* Enhanced user experience and efficiency by integrating **React** and **GraphQL**, optimizing data fetching.
* Created Postman collections for API testing, ensuring seamless integration and reliability.

**Environment:** Java 8, Spring Boot, Spring REST, Apache Kafka, Mockito, Swagger, Spring Cloud, Cassandra, MongoDB, NoSQL, Jenkins, Git.

**SKILLS**

| **Languages** | Java 8/11/17, Python, SQL, Javascript, Shell Scripting |
| --- | --- |
| **Frameworks** | Spring Boot, Spring MVC, Spring Security, GraphQL, OpenAPI, APIGEE, Hibernate |
| **Web development** | React.js, HTML, CSS, PHP, Postman |
| **DevOps Tools** | Jenkins, Git, GitLab, GitHub Actions, Maven, SonarQube, Swagger |
| **Cloud Platforms** | AWS (EC2, S3, Lambda, EKS, Glue, API Gateway, CloudWatch), Kubernetes |
| **Databases** | Oracle, PostgreSQL, MySQL, Cassandra, SQL Server |
| **Messaging/Streaming** | Apache Kafka, Apache Flink, JMS |
| **Monitoring & Logging** | ELK Stack (Elasticsearch, Logstash, Kibana), Grafana, Prometheus |
| **CI/CD & Containerization** | Docker, Kubernetes |
| **Data Processing** | AWS Glue, Azure Data Factory, Python, Pandas, PL/SQL |
| **Reporting & BI** | Tableau, Cognos |
| **Testing** | JUnit 4/5, Mockito, Spring Test, Selenium |
| **Version Control** | GitHub, GitLab, Bitbucket |
| **Security** | OAuth 2.0, PingFederate, Spring Security |
| **Certifications** | Google Associate Cloud Engineer |