Manoj Kumar

Professional Summary

Java Backend Developer with 6.5+ years of experience in building scalable backend systems using Java 8+, Spring Boot, REST APIs, Microservices, and Kafka. Skilled in multithreading, DB optimization, Agile delivery, and CI/CD automation using Jenkins + Docker. Hands-on with AWS EC2 deployments and Kafka-based async architectures.

Technical Skills

Languages: Java 8+, SQL
Core Concepts: Multithreading, Collections, DSA, Design Patterns
Frameworks: Spring Boot, Spring MVC, Hibernate
Architecture: REST APIs, Microservices, Event-driven systems
Messaging: Apache Kafka (Producer/Consumer, Topics, Event Contracts)
DevOps: Docker, Docker Compose, Jenkins, GitHub Webhook, CI/CD Pipelines
Cloud/Infra: AWS EC2, Linux (Ubuntu), SSH, Systemd, Shell Scripting
Databases: MySQL, Oracle
Testing: JUnit, Mockito
Developer Tools: Git, Maven, Postman, JIRA, Eclipse, STS, IntelliJ
Servers: Apache Tomcat, Oracle WebLogic

Professional Experience

Senior Technology Consultant Oct 2021 – May 2023
EY (Client: HSBC) — Home Mortgage Application *Bangalore*

- Built Spring Boot microservices for mortgage recommendations based on customer data
- Improved Oracle query performance, reducing response times by 30%
- Integrated external APIs for rate comparison and implemented caching for static datasets
- Participated in sprint planning, tech discussions, and code reviews in Agile setup
- Implemented Jenkins CI/CD pipelines to automate builds, Docker image creation, and microservice deployments across lower environments
- Containerized Spring Boot services using Docker and used Docker Compose for QA/UAT environments
- Managed multi-env configurations via Spring Profiles (‘dev‘, ‘uat‘, ‘prod‘) integrated with Docker containers

Technology Analyst Sep 2019 – Sep 2021
Infosys (Client: iChor) — Manufacturing Execution System (MES) *Bangalore*

- Built backend modules in Spring Boot to manage production work orders and transactions
- Utilized ExecutorService with a fixed thread pool to process multiple ESS job IDs in parallel, boosting throughput by 40%
- Created transactional REST APIs and complex Oracle procedures for high-volume updates
- Created Kafka producer-consumer pipeline to asynchronously update printer configurations and alert queues
- Used Docker Compose to locally spin up Spring Boot + MySQL services for end-to-end module testing
- Built a complete CI/CD pipeline to build, test, containerize, and deploy Spring Boot microservices automatically to Kubernetes
- Built Jenkins pipeline for Maven builds, Docker image creation, and automated deployments to staging servers
- Created shell scripts for container startup and automated service restarts using ‘systemd‘
- Defined ‘application.yml‘ profiles to cleanly separate test and production deployments

Senior Systems Engineer Oct 2016 – Sep 2019
TCS (Client: Bank of America) — Background Screening Investigations *Chennai, Bangalore*

- Developed REST APIs for background screening services using Spring Boot + Oracle WebLogic
- Designed efficient Oracle SQL queries to handle large-scale applicant data
- Conducted unit testing with JUnit/Mockito and participated in regular code reviews
- Built CI/CD pipelines in Jenkins for WAR builds, test execution, and deployment to Oracle WebLogic servers
- Explored containerizing internal services using Docker for local testing and reduced config drift
- Maintained environment-specific configurations using externalized property files and Spring Profiles

PERSONAL PROJECTS

Order Management System | *Microservices, Kafka, Docker, AWS, CI/CD* June 2023 – Present

- Designed an end-to-end microservices-based Order Management System from scratch using Spring Boot (Java 17)
- Developed modular services: order, inventory — fully decoupled via Kafka
- Used Apache Kafka (with DTO-based contracts) for async event flow across services via topics like ‘order-topic‘, ‘inventory-topic‘
- Built RESTful APIs for order placement, item stock check, and event-driven order confirmation logic
- Wrote 20+ unit tests using JUnit and Mockito covering services, controllers, and Kafka listeners
- Dockerized all services, MySQL DB, Kafka Zookeeper; managed via Docker Compose locally and on AWS EC2
- Pushed Docker images to DockerHub and pulled from EC2 during deployment
- Used SSH and shell scripts to deploy using Docker Compose and manage services via systemd
- Set up Spring profiles (‘application-local.yml‘, ‘application-aws.yml‘) for clean env-specific configuration
- Configured Jenkins pipeline with GitHub Webhook → Maven build → Docker build/push → SSH EC2 auto-deploy
- Deployed these microservices with Kafka, MySQL to AWS EC2 with CI/CD pipeline using Jenkins tool to reduced manual steps by 90%

EDUCATION

Techno India Salt Lake, WBUT Kolkata, West Bengal

Bachelor of Technology in Computer Science and Engineering, DGPA: 8.01 *July 2012 – June 2016*