Name: Anoohya Sri N
Email: anoohyan52@gmail.com | Phone No: +1((916)-382-2984


Professional Summary:

Experienced **Senior Java Backend Developer** with 12 years of expertise in designing, developing, and deploying robust, scalable, and high-performance backend applications and microservices. Proven track record of working with **Spring Boot**, **Hibernate**, and **RESTful APIs**, along with cloud-native solutions on **AWS** and **Azure**. Adept at implementing **event-driven architecture**, managing CI/CD pipelines, and optimizing system performance. Skilled in ensuring security compliance with standards such as **OWASP**, **PCI**, and **SOC2**.

- Extensive experience in building Java-based backend applications using Spring Boot and Hibernate.
- AWS Certified Senior Java Backend Developer with 12+ years of experience designing scalable microservices and API-first architectures using Spring Boot, AWS Cloud, and CI/CD pipelines in Agile environments.
- Proven ownership of on-call rotations (PagerDuty) and proactive risk escalation, ensuring <10 min MTTR for Tier-1 services.
- Skilled in developing event-driven, API-first systems using Docker, Kubernetes, Kafka, Terraform, and resilient cloud infrastructure with observability integrations such as Prometheus, ELK, and Datadog.
- Recently focused on Kotlin-based microservices, transitioning Java services into Kotlin for improved readability and performance in Spring Boot.
- Strong experience integrating AI modules and secure REST/SOAP services within distributed systems hosted on AWS and Kubernetes.
- Proficient in implementing Spring Cloud features like Config Server, Eureka, and load balanced Feign clients to manage microservice communication and configuration dynamically.
- Experienced in using Prometheus and Datadog alongside AWS CloudWatch and ELK stack to monitor, alert, and improve system observability and uptime.
- Hands-on experience building and integrating GraphQL APIs using Spring Boot and Netflix DGS, enabling flexible frontend data querying across services
- Recent experience deploying microservices on Google Cloud Platform (GKE, Cloud Spanner) with GitHub Actions and Argo CD for secure, scalable infrastructure automation
- Expertise in microservices architecture and domain-driven design (DDD) to ensure scalable, modular solutions.
- Skilled in designing and developing RESTful APIs, with strong knowledge of API versioning, OAuth 2.0, JWT, and authentication/authorization mechanisms.
- Proficient in AWS services, including EC2, Lambda, RDS, DynamoDB, S3, ECS, Fargate, EKS, SNS, SQS, API Gateway, and CloudWatch.
- Experience in designing, deploying, and managing cloud-native applications and services on Azure, including Azure Kubernetes Service (AKS), SQL Database, Cosmos DB, Event Hubs, and Key Vault.
- Expertise in implementing serverless functions using AWS Lambda to reduce infrastructure costs and improve scalability.
- Strong background in setting up CI/CD pipelines using Jenkins, GitHub Actions, Terraform, and AWS Code Pipeline.
- Proficient in automating infrastructure provisioning with Terraform and CloudFormation, ensuring repeatable, consistent environment deployments.
- Implemented OAuth2 and JWT for secure authentication and authorization across systems.
- Ensured data encryption (in transit and at rest) using AWS KMS and Azure Key Vault.

- Complied with industry standards like OWASP, PCI DSS, and SOC2, ensuring the integrity, security, and confidentiality of sensitive data.
- Optimized database performance using advanced indexing, query tuning, and Azure Cache for Redis to reduce latency and improve system throughput.
- Instrumented applications for observability using AWS CloudWatch, Elasticsearch, Kibana (ELK Stack), Grafana, and Azure Monitor for proactive monitoring, logging, and troubleshooting.
- Enhanced service resilience by implementing Resilience4j for circuit breakers, retries, and fallback mechanisms.
- Experience with relational databases such as Oracle, MSSQL, PostgreSQL, and Azure SQL Database.
- Implemented MongoDB and Cosmos DB for NoSQL data storage, ensuring flexibility and scalability.
- Designed and optimized database schemas for improved query efficiency and system responsiveness.
- Actively participated in Agile ceremonies, including sprint planning, backlog grooming, and standups, ensuring timely delivery of features.
- Collaborated with cross-functional teams using Jira, Confluence, and Slack for streamlined communication and project management.
- Extensive experience in unit testing with JUnit 5, Mockito, and EasyMock to ensure high code quality.
- Conducted integration testing using Test Containers and Azure Test Plans, ensuring that microservices interact seamlessly.
- Used SonarQube for continuous code quality checks.
- Adept at transforming legacy applications into modern microservices-based systems, enhancing scalability, performance, and maintainability.
- Recognized for delivering mission-critical solutions in highly regulated environments such as finance, retail, and e-commerce.

Technical Skills:

Programming Languages	Core Java, J2EE, Shell Scripting, Kotlin, python, Node.js, Go, Typescript
Backend Development Technologies	Spring Boot, Spring Cloud (Eureka, Config Server, Feign Client), Hibernate, JPA, RESTful APIs, SOAP, GraphQL, Microservices Architecture
Testing Tools	JUnit 5, Mockito, TDD, Easy Mock, Test Containers, Postman, Newman
API Standards	REST, SOAP, Swagger/OpenAI 3.0
Build Tools	Jenkins, Code Pipeline, Git, Bitbucket, Maven, SonarQube**,** Terraform, GitLab CI/CD
DevOps Tools	Docker, Kubernetes, GKE, Terraform, CloudFormation, Lucid Chart, Draw.io
Databases	Oracle, MSSQL, PostgreSQL, MySQL, MongoDB, DynamoDB, Azure SQL Database, Google Cloud Spanner

Messaging	Apache Kafka, JMS (Java Message Service), AWS SNS, AWS SQS, Azure Event Hubs, RabbitMQ
GenAI	Prompt Engineering, Retrieval-Augmented Generation (RAG), Vector DBs, LLM model basics
SDLC Methodologies	Agile, Waterfall, SCRUM.
Logging Tools	AWS CloudWatch, Elasticsearch, Prometheus, Datadog, Kibana, Grafana, Zipkin
Operating Systems	Windows, Linux and UNIX.
Version Control	GitHub, Git, IntelliJ
Security & Compliance	OAuth 2.0, JWT, PCI/SOC2 Compliance, OWASP Top 10
Cloud Platforms	AWS (API Gateway, ECS, Fargate, Lambda, RDS, S3, EC2, VPC, CloudWatch, ALB/NLB, Route 53, CloudFormation, IAM, SNS, SQS), Azure (AKS, SQL Database, Blob Storage, Cosmos DB, Key Vault, Event Hubs), GCP (GKE, Cloud Spanner)
Software Tools	Visual Studio Code (VSCode) and Microsoft Office, Jira, Confluence

Certifications:

- **AWS Certified Solution Architect – Associate (Link)**
- **Oracle Cloud Infrastructure 2023 Certified Application Integration Professional (Link)**
- **Oracle Cloud Infrastructure 2023 Certified Foundations Associate (Link)**
- **Certified in Version Control with Git (Link)**

Professional Experience:

USAA, San Antonio, TX **Jan 2022- Present**

Role: Senior Java Backend Developer

USAA, a leading financial services company serving military members and their families, focuses on delivering secure and innovative banking and insurance solutions. As a Senior Java Backend Developer, I led the design and deployment of Spring Boot microservices for Virtual Card Services, following DDD and 12-Factor App principles. I developed secure RESTful APIs with AWS API Gateway, using OAuth 2.0 and JWT for authentication. I utilized AWS SNS and SQS for asynchronous communication and containerized applications with Docker, AWS ECS, and Kubernetes (EKS). I built CI/CD pipelines with Jenkins, GitHub Actions, and Terraform, automated infrastructure with CloudFormation, and ensured observability using AWS CloudWatch and the ELK Stack. Additionally, I implemented Resilience4j for service resilience and ensured compliance with OWASP and PCI/SOC2 standards.

Responsibilities:

- Led the end-to-end design, implemented robust **Core Java** backend microservices using Spring Boot, applying object-oriented design principles for Virtual Card Services with strict adherence to **Domain-Driven Design (DDD)** and **12-Factor App** principles.
- Leveraged Java's multi-threading capabilities to develop asynchronous processing systems integrated with **AWS SNS** and **SQS,** ensuring high throughput and fault tolerance in mission-critical financial applications.
- Developed and exposed secure, scalable **RESTful APIs** integrated with **AWS API Gateway**, implementing **OAuth 2.0** and **JWT** for authentication and authorization.

- Benchmarked GitHub Actions vs GitLab CI/CD for internal CI pipeline; documented migration steps from Jenkins to GitLab for select modules.
- Integrated **Amazon ElastiCache (Redis)** to cache token-validation results, cutting p99 latency by 40 %.
- Applied **Kotlin DSL** for writing infrastructure as code with Terraform for environment-specific provisioning in AWS.
- Implemented **coroutine-based async service layers** for high-throughput card validation APIs, reducing thread overhead and improving latency.
- Integrated **GraphQL API endpoints** using Spring Boot and Netflix DGS framework for dynamic client-side querying and aggregation in fraud-prevention services.
- Conducted internal knowledge sharing on transitioning from REST to **GraphQL APIs** using schema stitching and federation techniques.
- Migrated legacy Java microservices to Kotlin using Spring Boot 3.x for Virtual Card Services.
- Designed and coded the XML parser to create the initial setup for the entire Java SWING GUI.
- Developed and deployed RESTful and SOAP-based services for cross-system integrations ensuring backward compatibility.
- Integrated MySQL-backed legacy components into Spring Boot microservices for audit and reference reconciliation purposes.
- Secured sensitive configurations and credentials using **AWS Secrets Manager** and implemented least-privilege access with **AWS IAM** roles and policies.
- Integrated Spring Cloud Config Server and Eureka for dynamic configuration and service discovery across microservices.
- Instrumented backend services using Prometheus and integrated with Grafana dashboards to provide real-time performance metrics alongside CloudWatch.
- Designed containerized backend applications with **Docker,** and deployed them using **AWS ECS, AWS Fargate,** and optionally **Kubernetes (EKS**) for orchestrated, high-availability deployments.
- Proven leader in backend modernization projects, with hands-on expertise in cloud-native deployments and security-compliant architectures in regulated industries.
- Demonstrated ability to guide cross-functional teams across multiple cloud platforms (AWS, Azure), ensuring scalability, security, and operational efficiency.
- Managed source control, branching strategies, and pull request workflows using **GitHub,** ensuring clean code integration through CI/CD pipelines.
- Built and optimized CI/CD pipelines utilizing **Jenkins**, **GitHub Actions, Terraform**, and **AWS Code Pipeline** for automated builds, security scans, testing, and deployments.
- Automated infrastructure provisioning with **Terraform** and **AWS CloudFormation,** supporting scalable and consistent environment deployments.
- Instrumented applications for observability using **AWS CloudWatch Logs, Custom Metrics, Elasticsearch,** and **Kibana (ELK Stack)** for proactive system monitoring and alerting.
- Implemented unit and integration testing with **JUnit 5, Mockito**, and **Easy Mock**, and developed containerized test environments using **Test Containers** for realistic integration tests.
- Enhanced service resilience by introducing circuit breakers, retry, and fallback mechanisms using **Resilience4j.**
- Integrated with React frontend teams, providing optimized RESTful endpoints and maintaining API contracts using Swagger.
- Contributed to prompt engineering POCs for internal fraud-detection tools using vector similarity queries and basic RAG techniques.

- Actively participated in Agile ceremonies, including sprint planning, backlog grooming, and daily standups, collaborating closely with cross-functional teams via **Jira, Confluence,** and **Slack.**
- Conducted performance tuning by analyzing **Heap Dumps, Thread Dumps**, and configuring optimized **Garbage Collection (G1GC)** settings for JVM applications.
- Ensured application and infrastructure compliance with **OWASP Top 10** security guidelines and internal audit controls for PCI/SOC2 standards.
- Collaborated with DevOps and Site Reliability Engineers to manage global traffic distribution using **AWS ALB/NLB, Route 53,** and **CloudFront.**

Environment: Core Java (Java 11/17), Spring Boot (3.1), Kotlin, RESTful APIs, OAuth 2.0, JWT, Node.js, AWS (API Gateway, SNS, SQS, Secrets Manager, IAM, ECS, Fargate, CloudWatch, ALB/NLB, Route 53, CloudFront, CloudFormation), MySQL, GitLab CI/CD, Docker, ElastiCache, SOAP, TDD, Kubernetes (EKS), GraphQL, GKE, Google Cloud Spanner, GitHub, Jenkins, Prometheus, Datadog, Spring Cloud (Eureka, Config Server, Feign), GitHub Actions, React Integration, Prompt Engineering, Vector DB, Swing, Terraform, AWS Code Pipeline, Elasticsearch, Kibana (ELK Stack), JUnit 5, Mockito, Easy Mock, Test Containers, Resilience4j, Jira, Confluence, Slack, G1GC, OWASP Top 10, PCI/SOC2 compliance.

Amazon, Sunnyvale, CA **May 2020 - Dec 2021**

Role: Senior Java Backend Developer

At Amazon, a global leader in e-commerce and cloud computing, I contributed to a major project aimed at enhancing customer-facing platforms. As a Senior Java Backend Developer, I developed scalable Java microservices using Spring Boot and Hibernate for high-traffic platforms, integrating Amazon RDS and DynamoDB**.** I containerized applications with Docker**,** AWS ECS**,** and Fargate, and automated CI/CD pipelines using AWS CodePipeline**,** Jenkins, and Terraform. I optimized performance with Elastic Cache and AWS Lambda**,** designed RESTful APIs with Swagger/OpenAPI 3.0, and implemented Kafka and JMS for asynchronous messaging**.** I ensured system observability with AWS CloudWatch, Grafana and maintained security with IAM policies and encryption**.**

Responsibilities:

- Developed scalable **Core Java** based microservices using **Spring Boot** and **Hibernate** to support high-traffic e-commerce platforms, ensuring fault tolerance and low latency.
- Implemented effective Java exception handling mechanisms and designed resilient failure recovery strategies to maintain system stability across distributed microservices architecture.
- Design the GUI for the Accounts and Purchase modules using Java SWING.
- Integrated **Amazon RDS** and **DynamoDB** to ensure efficient data storage and real-time querying with high availability and reliability.
- Designed Kotlin-based microservices for checkout session tracking, improving latency and readability over legacy Java.
- Participated in TDD cycles using JUnit5 and Mockito, achieving 90%+ code coverage and fewer post-deploy defects.
- Built and integrated RESTful and GraphQL web services using JAX-RS, enhancing communication across distributed systems and ensuring smooth interaction between client and server.
- Implemented containerized applications using **Docker**, orchestrated with **AWS ECS** and **Fargate** for improved scalability and CI/CD pipeline automation.
- Acted as escalation lead during **PagerDuty** on-call; reduced mean time-to-engage by 30 %.
- Enhanced microservices performance by utilizing **Elastic Cache** for caching and optimizing database calls to reduce response times.
- Adopted **Kotlin Coroutine Flows** in Spring Boot to handle real-time inventory updates, improving response times and throughput during flash sales.

- Designed **unit and integration tests using Kotlin + JUnit5**, adhering to TDD practices for feature additions and regressions.
- Led migration proof-of-concept from AWS DynamoDB to **Google Cloud Spanner**, leveraging its PostgreSQL dialect for internal benchmarking.
- Design and develop the GUI screens for the application using Java Swing.
- Deployed containerized workloads on **Google Kubernetes Engine (GKE)** for hybrid cloud architecture evaluations.
- Implemented Prometheus exporters for JVM metrics and integrated with Grafana dashboards for internal service health visualization.
- Extended Terraform templates for hybrid deployments involving legacy AWS EC2 and new Lambda + ECS stacks.
- Created serverless functions with **AWS Lambda** to optimize backend operations and reduce infrastructure costs while ensuring scalability.
- Refactored microservices to use Spring Cloud LoadBalancer and centralized configuration using Spring Cloud Config.
- Benchmarked GitHub Actions vs GitLab CI/CD for internal deployment pipelines and documented trade-offs for team adoption.
- Designed and implemented **RESTful APIs** using **Swagger/OpenAPI 3.0** standards, ensuring easy integration and consistent API documentation for front-end and third-party applications.
- Collaborated with Node.js-based microservices handling checkout-session verification logic, ensuring seamless inter-service communication via AWS SQS and Kafka.
- Designed and developed the GUI using Java swing components for new modules
- Enhanced API compliance by integrating REST and legacy SOAP endpoints with shared schemas, assisting API-first integration with external vendors.
- Managed deployments and automated rollbacks using **AWS CodePipeline, Jenkins,** and **Terraform,** ensuring efficient deployment pipelines and infrastructure as code.
- Collaborated with TypeScript-based frontend teams to integrate microservices using OpenAPI 3.0 specifications.
- Developed Lambda-based utilities in Python to support logging and alerting workflows for internal DevOps teams.
- Integrated **Kafka** and **JMS** messaging systems to enable asynchronous communication between microservices for decoupled and reliable service interactions.
- Set up real-time application monitoring and alerting using **AWS CloudWatch** and **Grafana dashboards**, improving system visibility and issue detection.
- Actively contributed to DevOps practices, optimizing the CI/CD pipeline and cloud infrastructure to ensure continuous and stable deployments across both **on-premises** and **AWS environments**.
- Collaborated with cross-functional teams to deliver customer-facing features, ensuring alignment with business goals and improving overall user experience.
- Implemented security best practices including encryption of sensitive data at rest and in transit, and **IAM policies** for fine-grained access control.
- Performed load balancing and traffic management using **AWS ELB** to ensure high availability and optimal performance of microservices across multiple availability zones.

Environment: Core Java (8), Spring Boot (2.3), Docker (19), Hibernate, Kotlin, SOAP, TDD, Amazon RDS, DynamoDB, AWS ECS, Fargate, CI/CD (Jenkins, Terraform, Pager Duty, AWS CodePipeline), Spring Cloud Config, GitLab CI/CD (comparison), Google Cloud Spanner, GKE, GraphQL, TypeScript integration, Node.js, REST/SOAP, messaging systems, Swing, Python Lambda Scripts, Elastic Cache, AWS Lambda, Swagger/OpenAPI 3.0, Kafka, JMS, AWS CloudWatch, Prometheus, Grafana, AWS ELB.

Home Depot, Atlanta, GA **Mar 2018- Apr 2020**

Role: Java Backend Developer

The Home Depot**,** a leading home improvement retailer, operates an extensive digital platform serving millions of customers. During this 2-year project, I led the migration of legacy backend systems to Azure Cloud, converting them into Spring Boot microservices deployed on Azure Kubernetes Service (AKS). I integrated Azure SQL Database, Cosmos DB, and PostgreSQL for reliable data storage and used Azure Active Directory (AAD) with OAuth2 and JWT for secure authentication. I developed RESTful APIs, integrated with Azure Blob Storage, and optimized performance with Azure Cache for Redis. CI/CD pipelines were automated using Azure DevOps, and real-time monitoring was enabled via Azure Monitor and Application Insights. Event-driven communication was implemented with Apache Kafka and Azure Event Hubs, ensuring a scalable, secure, and high-performance system.

Responsibilities:

- Led the migration of legacy backend systems to **Azure Cloud**, transforming them into cloud-native **Spring Boot** microservices deployed on **Azure Kubernetes Service (AKS).**
- Applied Java dependency injection patterns and Spring Boot integration create modular, testable code that seamlessly integrated with Azure cloud services.
- Integrated **Azure SQL Database**, **Cosmos DB**, and **PostgreSQL** for scalable and reliable data storage, ensuring high availability and performance.
- Implemented secure authentication and authorization mechanisms using **Spring Security**, **OAuth2**, **JWT**, and **Azure Active Directory (AAD)** for user management and role-based access control.
- Instituted rotating **PagerDuty** schedule and Jenkins slack alerts.
- Lead the team for building the Java Swing GUI tools, SQLOne console and Migration report generator
- Designed and implemented Spring Cloud Gateway for internal API routing, rate limiting, and centralized authorization.
- Developed AI-driven chatbot integrations using Azure OpenAI for automated helpdesk responses with RAG-based document support.
- Built Kotlin-based services for **AI chatbot backend integrations** using Azure OpenAI, enabling RAG-based helpdesk automation.
- Used **Kotlin with Spring Cloud Gateway** for building internal API proxies with built-in rate limiting and circuit breaking.
- Integrated **Kotlin Gradle scripts** for managing multi-module microservices builds and improving developer productivity.
- Led SOAP service integration with legacy inventory systems to bridge older retail platforms with microservice APIs.
- Built custom infrastructure-as-code modules using **Terraform** to provision Azure SQL, AKS, and Cosmos DB with role-based access and monitoring
- Integrated **GitLab CI/CD** for select feature teams as part of DevOps streamlining.
- Evaluated RabbitMQ as an alternative to Kafka for specific microservices that required low latency guaranteed delivery.
- Designed and developed RESTful APIs and microservices that interfaced with **Azure Blob Storage** for efficient file storage and retrieval.
- Built and maintained **CI/CD pipelines** using **Azure DevOps**, integrating automated builds, tests, and deployments to ensure fast and reliable delivery.

- Developed internal Node.js scripts to assist with Azure DevOps CI validation checks, enabling pre-deployment API endpoint verification.
- Implemented fine-grained RBAC using Spring Security and Azure OAuth2 integrations, achieving compliance with enterprise identity governance policies.
- Contributed to internal LLM-based chatbot integration by exploring use of Azure OpenAI for employee helpdesk automation.
- Built TypeScript-compatible backend services and ensured CORS/API integration with planned React UIs.
- Deployed containerized applications using **Docker** and orchestrated them with **Azure Kubernetes Service (AKS)** to enhance scalability and availability.
- Enhanced real-time data processing with **Apache Kafka** and integrated **Azure Event Hubs** for event-driven microservices communication.
- Writing various custom swing components using JTable, JTree, etc
- Utilized **Azure Key Vault** for secure storage and management of sensitive configuration data and credentials.
- Leveraged **Azure Monitor** and **Azure Application Insights** for real-time application monitoring, logging, and troubleshooting.
- Implemented automated performance and security tests using **Azure Test Plans** and **SonarQube** integration to maintain high code quality and system reliability.
- Optimized database performance using **Azure Cache for Redis** to reduce latency and improve throughput in high-traffic applications.

Environment: Core Java (8), Spring Boot (2.2), Azure Kubernetes Service (AKS), Azure SQL Database, React-compatible APIs, Azure OpenAI/GenAI, Cosmos DB, Spring Cloud Gateway, RabbitMQ, PostgreSQL, Azure Active Directory (AAD), RAG, OAuth2, JWT, ElasticCache, Docker, Terraform (Azure), SOAP, GitLab CI/CD, Azure DevOps, Azure Blob Storage, Apache Kafka, Azure Event Hubs, Swing, Azure Key Vault, Node.js, OAuth 2.0, CI/CD support, Azure Monitor, Azure Application Insights, Azure Test Plans, SonarQube, Azure Cache for Redis.

New York State - Information of Technology services Department of Office of Mental Health
Albany, NY **Jan 2017- Feb 2018**
Role: Senior Java Backend Engineer

During my tenure at the New York State - Information of Technology Services, Department of Office of Mental Health, I developed and deployed Spring Boot microservices on AWS EC2 and AWS EKS for a healthcare platform, integrating MongoDB for scalable, flexible data storage. I optimized system performance using AWS Lambda for event-driven processing and AWS S3 for scalable storage. I implemented AWS API Gateway for secure API access and used AWS CloudFormation and Terraform for infrastructure automation. Monitoring was set up with AWS CloudWatch, and event-driven architecture was powered by AWS SQS and SNS. I automated deployments with Jenkins and AWS Code Pipeline to ensure continuous integration and rapid deployment.

Responsibilities:

- Developed **core Java** solutions and **Spring Boot** microservices that incorporated industry-standard design patterns (Factory, Singleton, Observer) for healthcare data processing systems deployed on **AWS EC2** and **EKS.**
- Created custom Java serialization/deserialization implementations to efficiently process healthcare data between services and **MongoDB** persistence layer.
- Designed and implemented **AWS Lambda** functions for event-driven processing and to improve system performance and reduce costs.

- Enabled low-latency session caching with **ElastiCache-Redis** in EKS.
- Led after-hours **PagerDuty** support for HIPAA-critical endpoints.
- Maintained and consumed both RESTful and SOAP web services for internal healthcare systems.
- Utilized **AWS S3** for scalable storage solutions to manage large volumes of healthcare-related data.
- Enabled inter-service communication through RabbitMQ for selected health data microservices, improving throughput and decoupling.
- Enabled advanced service metrics using **Prometheus Node Exporters** and integrated into **Grafana dashboards** for real-time operational insight.
- Implemented **MySQL** as a migration target from legacy Oracle schemas for select health record subsystems.
- Monitored infrastructure and app health with Datadog integrations alongside AWS-native CloudWatch.
- Implemented **AWS API Gateway** to manage **RESTful APIs** and ensure secure, scalable access to backend services.
- Employed **AWS CloudFormation** and **Terraform** for infrastructure automation and consistent, repeatable cloud environment provisioning.
- Participated in early exploration of code generation tools (Amazon Code Whisperer) to accelerate Lambda and Terraform scripting.
- Worked on an internal GenAI knowledge retrieval system using vector-based search indexing via Elasticsearch and MongoDB.
- Integrated **AWS CloudWatch** for centralized logging and real-time monitoring to track system health and performance metrics.
- Designed **AWS SQS** and **SNS** for decoupling services and building event-driven, fault-tolerant architectures.
- Used **AWS IAM** for fine-grained access control and ensuring that only authorized services and users can access sensitive healthcare data.
- Optimized data processing workflows by using **AWS Step Functions** for orchestrating multi-step processes with error handling and retries.
- Applied **AWS KMS** for data encryption at rest and in transit to ensure the confidentiality and integrity of sensitive healthcare data.
- Automated testing and deployment pipelines using **Jenkins**, **AWS Code Build**, and **AWS Code Pipeline**, reducing deployment time and ensuring continuous integration.

Environment: Core Java (8), Spring Boot, MongoDB, AWS EC2, AWS EKS, MongoDB, AWS Lambda, AWS S3, AWS API Gateway, AWS CloudFormation, RabbitMQ, Datadog, Terraform, TDD,Prometheus, SOAP, Grafana, MySQL, Code Whisperer, Vector Indexing with Elasticsearch, AWS CloudWatch, AWS SQS, AWS SNS, AWS IAM, AWS KMS, AWS Step Functions, Jenkins, Swing, AWS Code Build, AWS Code Pipeline.

Tech Mahindra, Hyderabad, India **May 2013 – Dec 2016**

Role: Junior Java Backend Developer

Worked as a Junior **Java Backend Developer** on a large-scale digital transformation project for a global retail and e-commerce client. The project focused on developing Java-based backend applications using Spring MVC and Hibernate for enterprise system management. Optimized Oracle and MSSQL databases for improved performance, and built REST APIs integrating with AWS EC2, Lambda, and S3. Automated builds with Maven and managed CI/CD pipelines using Jenkins. Worked in an Agile environment, improving system functionality and ensuring seamless collaboration with cross-functional teams.

Responsibilities:

- Developed and maintained core Java based backend applications using **Spring MVC** and **Hibernate** for efficient enterprise system management.
- Designed and optimized **Oracle** and **MSSQL** database schemas for high-performance data storage, improving query efficiency and overall system responsiveness.
- Built **REST APIs** to integrate backend systems with **AWS EC2, Lambda,** and **S3** for cloud-native application enhancements.
- Produced block-level deployment diagrams for client sign-off.
- Automated build processes with **Maven**, ensuring consistency across different development and production environments.
- Participated in evaluation of GitLab CI/CD pipelines during DevOps onboarding phase before Jenkins standardization.
- Assisted in implementing GitLab CI/CD pipelines to speed up backend deployments before GitHub migration.
- Managed deployments via **Jenkins,** establishing automated CI/CD pipelines for streamlined code delivery, testing, and deployment.
- Worked with cross-functional teams to improve backend functionality, integrating various **Oracle** and **MSSQL** database-driven solutions.
- Optimized database performance through indexing, query optimization, and tuning for large-scale applications.
- Supported version control using **Git**, ensuring proper branch management and seamless collaboration between team members.
- Collaborated in a fast-paced Agile environment, participating in sprint planning, daily standups, and retrospective meetings.

Environment: Java JDK - 1.4/5/6/7, Hibernate, GitLab CI/CD, Spring MVC, Oracle, MSSQL, REST APIs, AWS EC2, AWS Lambda, AWS S3, Maven, Jenkins, CI/CD, Git, Agile.