Vamsi Krishna Bodaballa

bvkrishna2000@gmail.com | +1 (947)-300-6432 | Github-Vamsi767| Portfolio | Detroit, Michigan

SUMMARY

Computer Science graduate with hands-on experience in building backend services, RESTful APIs, and full-stack web applications using Java, Spring Boot, and Angular/React. Worked on cloud platforms like AWS and GCP, focusing on microservices, automation, and scalable system design.

Experienced in writing clean, testable, and maintainable code using object-oriented principles. Skilled in building CI/CD pipelines, working with Docker, Kubernetes, and Kafka, and writing automation scripts for testing and deployment.

Comfortable working in Agile teams, solving real-world problems, and delivering high-quality code. Enjoys creating new features, debugging issues, and contributing to fast-paced projects that impact users at scale.

Contribute to personal and open-source projects on GitHub while continuously learning through technical courses and certifications to strengthen skills in cloud computing, automation, and software engineering.

EDUCATION

University of Kansas, Lawrence, KS, USA **Aug 2023 – May 2025**
Master of Science: Computer Science. CGPA: 3.89/4.00
Courses: Data Structures and Algorithms, Machine Learning, Deep Learning, Computer Vision.

PROFESSIONAL EXPERIENCE

Java Full Stack Developer – Client: General Motors, MI, USA **June 2024 - Present**

- Designed and deployed scalable, highly available architectures on **AWS**, resulting in a **25%** improvement in system reliability.
- Built and deployed microservices using **Java**, **Python**, **Spring Boot**, and **Spring Security**, enabling scalable apps within cloud.
- Designed **RESTful APIs** and integrated them into Angular apps, and improving frontend responsiveness.
- Developed dynamic, reusable UI components using **Angular 9, HTML, CSS**, and **TypeScript** to support internal platform tools.
- Automated **infrastructure provisioning** with Terraform and AWS CloudFormation, reducing manual setup time by 20%.
- Automated Kubernetes (**EKS**) cluster deployments, upgrades, and workload management using Terraform and Ansible.
- Implemented **Kafka**-based event pipelines for asynchronous messaging, reducing coupling and improving system resiliency.
- Modeled NoSQL data and used **Redis** for caching strategies, lowering read latency and enhancing query performance.
- Optimized SQL queries for **MySQL** to support real-time transactions and data consistency across distributed modules.
- Containerized backend services using **Docker** and deployed them on **Kubernetes**, achieving consistent, scalable environments.
- Created scheduled job workflows using **Spring Batch** and **Quartz**, automating daily tasks and long-running processes.
- Modularized logging and error handling with **Spring AOP**, improving observability while keeping codebase clean.
- Wrote unit tests in **JUnit** and **Mockito** for service-layer logic, increasing code coverage and preventing regressions.
- Built integration **test** cases to validate system workflows, collaborating with **QA** to ensure test completeness.
- Automated **UI tests** using **Jasmine** and **Protractor**, covering major user paths and improving UI test reliability.
- Streamlined deployments by implementing **CI/CD pipelines** with Code Pipeline and Jenkins, cutting deployment time by 30%.
- Participated in test plan reviews and triaged defects, improving test cycles and coordinating across QA teams.
- Applied object-oriented principles to build maintainable and testable modules across the application lifecycle.
- Managed Linux-based environments (Ubuntu and Amazon Linux) for AWS system optimization, troubleshooting, configuration.
- Strong understanding of Linux file systems, disk space management, memory and CPU performance, security best practices.
- Ensured **24x7 cloud support** by actively monitoring and resolving performance issues with Prometheus and AWS CloudWatch.
- Developed monitoring and alerting solutions with **Splunk**, achieving a **40% reduction** in Mean Time to Recovery (MTTR).
- Conducted peer code reviews and shared reusable patterns during team syncs to deliver stable features under tight timelines.

Environment: Java, Python, Spring Boot, Spring Security, Angular, MySQL, DynamoDB, Redis, Kafka, Bash, Linux, Terraform, Kubernetes, Jenkins, Jasmine, Protractor, JUnit, Mockito, GCP, AWS, Git, OAuth 2.0, REST APIs, Networking, AWS Services (EKS, Lambda, EC2, S3, IAM, API Gateway, Code Pipeline, CloudFormation, Cloud Front, VPC, CloudFront, CloudWatch, CloudTrail, KMS, Security Groups), Git, GitHub.

Software Developer in Test – Walmart. **Jul 2021 – Jul 2023**

- Automated tasks in a **Linux**-based environment **(RHEL, Ubuntu)** using Python and Bash by **reducing manual setup time by 20%.**
- Collaborated with cross-functional teams to design and support Walmart's item ingestion pipeline for supplier product data.
- Developed backend microservices using **Python, Java** to process large-scale supplier data with integrated logging and monitoring.
- Built interactive internal dashboards using **Angular and TypeScript** to visualize feed metrics, failures, and real-time processing status.
- Used **Kafka** for stream processing and integrated it with **AWS SNS/SQS** for asynchronous communication and notification flows.
- Performed **REST API** testing using **Postman**, validating response codes, payload structures, and schema compatibility across services.
- Wrote SQL and **PL/SQL** queries in **PostgreSQL** and **Oracle** to verify data pipelines and troubleshoot complex ingestion errors.
- Utilized **AWS Lambda, CloudWatch, and IAM** to automate error handling, monitoring, and secure access control across services.
- Used ServiceNow (**SNOW**) to log, prioritize, and manage incidents, ensuring SLAs were met and issue trends were documented.
- Integrated **F5 GTM/LTM** for high availability and global traffic distribution across production and staging environments.
- Maintained Jenkins-based **CI/CD pipelines**, containerized with **Docker**, and deployed them via **Amazon EKS and Elastic Beanstalk**.

- Built test suites with **JUnit** and **Mockito** to validate service logic and integrated them into Jenkins pipelines for automated execution.
- Maintained **QA** documentation in **Confluence** and used **Jira** for managing sprints, user stories, and defect resolution tracking.
- Configured environment health monitoring using **Grafana and Kibana**, and setup **Slack alerts** for proactive incident notifications.
- Applied **OOP** principles to develop scalable, reusable modules and collaborated on peer code reviews and knowledge sharing.
- Contributed to exploratory, regression, and smoke testing during sprint cycles and production rollouts.
- Mentored L1 QA members and guided test strategy design and conducted root cause analysis for flaky tests with SRE teams.
- Validated deployment readiness post-release and provided signoffs after production sanity checks and rollback verification.

Environment: Linux (RHEL, Ubuntu), Bash Scripting, Python, Java, Spring Boot, Angular, Kafka, PostgreSQL, Oracle, Redis, Docker, Jenkins, AWS (SNS, SQS, Lambda, IAM, EKS, CloudWatch, Elastic Beanstalk), F5 GTM/LTM, ServiceNow, JUnit, Mockito, Postman, Confluence, Jira, Grafana, Kibana, REST APIs

INTERNSHIP/OTHER EXPERIENCE

SOFTWARE DEVELOPER INTERN – Dovekie Technologies **Jan 2021 – Jun 2021**

- Worked on web application development using **Java 8**, Spring Boot, and **Thymeleaf**, deploying apps on **Apache Tomcat servers**.
- Built interactive front-end interfaces using Angular 9 and **Bootstrap**, ensuring responsiveness and cross-browser compatibility.
- Developed backend RESTful APIs and business logic for pharmaceutical systems, improving system throughput by 20%.
- Used **Hibernate ORM** for database access and optimized SQL queries in **PostgreSQL** to reduce execution time and created batch jobs using Spring Batch and automated job scheduling for data processing using **Quartz Scheduler**.
- Utilized Docker for containerizing applications, achieving consistency across dev, test, and production environments.
- Developed and deployed microservices on Google Kubernetes Engine (**GKE**), enabling horizontal scaling and load balancing.
- Wrote **Python** and **Bash** scripts on **Linux** for automating deployment tasks and service monitoring across staging environments.
- Validated API functionality and data transformations using **Postman**, **JUnit**, and **TestNG** for robust backend testing.
- Managed builds using **Maven** and **Jenkins**, streamlining CI/CD workflows and improving deployment efficiency.
- Participated in test case design and helped QA validate regression, smoke, and exploratory test scenarios.
- Used Git for version control and collaborated through GitHub pull requests and code reviews.
- Assisted in tuning application and DB performance by monitoring logs, analyzing metrics, and applying fixes.

Environment: Java, Spring Boot, Thymeleaf, Angular, PostgreSQL, Docker, Kubernetes (GKE), Jenkins, Python, Bash, Linux, Maven, Git, GitHub, Postman, JUnit, TestNG, Jira, Confluence, REST APIs

Red Hat Developer Member – Red Hat **Jan 2025 – Present**

- Shared technical solutions, best practices, and insights in community forums and group discussions on open source & cloud technologies.
- Facilitated knowledge-sharing sessions on open-source tools and practices, promoting continuous learning among local IT professionals.

IT Intern – BSNL **Dec 2018 – Dec 2018**

- Assisted in implementing ITIL-based practices for application hosting and infrastructure review, contributing to system quality.
- Conducted user training sessions to raise awareness of cybersecurity hygiene, helping reduce common security incidents by 20%.

SKILLS

Programming Languages:	Java, Python, TypeScript, Golang, Shell Scripting, C++, YAML
Cloud Platforms:	Amazon Web Services, Google Cloud Platform, Microsoft Azure
Database & Frameworks:	SQL, MySQL, PostgreSQL, NoSQL, DynamoDB, Django, Flask, Spring Boot, Node.js, React.
AIOps & LLM:	Gen AI, Vertex AI, Amazon Bedrock, SageMaker, GenAI APIs, Ollama, Docker Runner.
Infrastructure Management:	Terraform, Ansible, CloudFormation, GCP Deployment Manager
Containerization:	Docker, Kubernetes, OpenShift, Helm, Istio
DevOps and CI CD:	Ansible, Jenkins, ArgoCD, CodePipeline, Maven, Gradle,Terraform, Docker, Kubernetes, Helm
Monitoring:	Prometheus, Grafana, ELK Stack, Datadog, AWS Cloud Watch, Splunk
QA & Testing:	JUnit, Mockito, Postman, SoapUI, JMeter, Jasmine, Protractor
Networking & Security:	VPC, VPN, Firewalls, Route 53, DNS, TCP/IP, OSI Layers, AWS GuardDuty, SAST, DAST
Web Services:	REST, SOAP, WSDL, JAX-RS, JAX-WS, JAXB, Swagger/OpenAPI
Operating Systems:	Unix/Linux (Ubuntu, Centos, Amazon Linux, RHEL), Windows.
Messaging & Streaming:	Kafka, RabbitMQ, JMS, AWS SNS/SQS, ActiveMQ
Project & Collaboration Tools:	Jira, Confluence, ServiceNow, Rally, Git, GitHub, WinSCP, PuTTY
Servers & IDE's:	Apache Tomcat, WebLogic, WebSphere, JBoss, IntelliJ, Eclipse, STS, VSCode, NetBeans
Other Skills/Tools:	Git, GitHub, ServiceNow, Agile and Scrum Methodologies (Jira, Confluence).

CERTIFICATIONS

- **AWS Certified Developer – Associate**
- **AWS Certified Cloud Practitioner**
- **Google Cloud Certified – Associate Cloud Engineer**

PROJECTS

Web Application with AWS Deployment | AWS, Tomcat, RabbitMQ, MySQL, Memcached, Route 53 |

- Designed and deployed a cloud-based web application with secure user sign-in and a dashboard to display backend service status.
- Configured GoDaddy DNS with AWS Route 53 to route HTTPS traffic to an Application Load Balancer secured by security groups.
- Deployed auto-scaling Tomcat instances behind the ALB to handle dynamic user load and ensure high availability.
- Integrated RabbitMQ for message queuing, Memcached for caching, and MySQL for persistent storage, each within private subnets secured via AWS security groups.
- Automated provisioning using Vagrant and shell scripts for consistent multi-service deployment across environments.

Source Code: **https://github.com/Vamsi-767/Scalable-Web-Application-with-AWS-Deployment**

Full-Stack E-Commerce Web Application | React.js, Spring Boot, MySQL, Java |

- Developed a fully functional e-commerce website using React.js for the frontend and Spring Boot for the backend.
- Implemented a user signup system, storing user information in a relational database for future interactions.
- Designed a product listing and cart system, allowing users to browse products, add items to the cart, and proceed to checkout.
- Created a RESTful API using Spring Boot to handle user registration, product management, and order processing.
- Used MySQL to store user details, cart items, and orders, ensuring data persistence and retrieval.
- Designed and implemented REST APIs for handling asset management, data validation, and processing automation.

Source Code: **https://github.com/Vamsi-767/Ecommerce-site**

Automated API Testing Framework | Python, Pytest, Postman, Jenkins, Splunk|

- Designed and developed a backend API testing framework as a personal project to enhance automation and validation skills.
- Created detailed test plans to validate functionality, error handling, and performance of RESTful APIs.
- Automated test scripts using Python and Pytest, achieving faster and repeatable validation cycles.
- Integrated test execution into Jenkins CI pipelines, enabling continuous feedback on system health.
- Conducted manual exploratory testing using Postman to detect edge case failures early in the development cycle.

Source Code: **Automated API Testing FrameWork (GitHub)**

Generative AI Assistant – GCP | Python, Vertex AI, Cloud Run, GitHub Actions|

- Built a Python-based web application that integrates with Google Vertex AI to generate user-specific content using LLMs.
- Deployed the service on Cloud Run with automated CI/CD pipelines via GitHub Actions for seamless delivery.
- Designed secure API access and implemented request logging for traceability and audit compliance.
- The system achieved 99.9% uptime, handled 100+ daily inference requests, and enabled scalable, cost-effective AI deployment.

Real-Time Serverless API Platform – AWS | Python, Lambda, DynamoDB, API Gateway|

- Designed and deployed a serverless application using AWS Lambda and API Gateway to handle real-time user interactions with ~3-second response latency.
- Integrated DynamoDB for fast, scalable data storage and retrieval, and applied Python-based logic to manage dynamic workflows.
- Eliminated the need for server provisioning, reducing operational costs by 50% and improving application scalability and resilience.