**Bala Subramanyam Duggirala**

**Full Stack Java Developer**

Email: bduggirala2@huskers.unl.edu | Mobile: +1 979‑422‑5067

LinkedIn: linkedin.com/in/balasubramanyamd | GitHub: github.com/Subramanyam6 | AI Research Group: iamas.unl.edu | Portfolio: [Bala SE Portfolio Website](https://polite-plant-0ce0f1c0f.6.azurestaticapps.net/)

**Professional Summary**

* Full‑stack engineer with 6+ years designing and shipping cloud‑native Java microservices and event‑driven systems on AWS and GCP.
* Expert in Java 11‑17, J2EE (Servlets, JSP, JMS), Spring Boot 3.x, Spring Cloud Netflix (Eureka, Feign, Resilience4j), and RESTful API design.
* Designed MongoDB + Oracle dual‑store architectures, optimizing query latency by 40% and supporting 3 M requests/day at p95 < 100 ms.
* Enabled 90%+ code coverage via JUnit 5, Mockito, pytest, and automated CI/CD pipelines (Jenkins, Cloud Build, GitHub Actions).
* Led cross‑site Scrum teams (US–India), mentored junior engineers, and executed blue/green Kubernetes deployments with zero downtime.

**Technical Skills**

|  |  |
| --- | --- |
| **Category** | **Technologies / Tools** |
| **Methodology** | SDLC, Agile Scrum, Sprint & Release Planning |
| **Languages** | Java 11‑17, Python, SQL, JavaScript/TypeScript, C, C++ |
| **Frameworks / Libraries** | Spring Boot 3, Spring Cloud Netflix (Eureka, Feign, Resilience4j), J2EE (Servlets, JSP, JMS), React.js, Flask, Django, Leaflet.js, D3.js, RESTful API |
| **IDEs** | IntelliJ IDEA, PyCharm, VS Code |
| **Spring Boot & Microservices** | Spring MVC, Spring Cloud Netflix (Eureka Server, Feign Client, Resilience4j), Kafka, RestTemplate, JUnit, Mockito |
| **Web Technologies** | HTML5, CSS3, Tailwind CSS, Bootstrap, JavaScript (ES Next), JSON, YAML, AJAX, WebSockets |
| **Testing Tools** | JUnit 5, Mockito, pytest, Cucumber, Locust, Log4j |
| **Application Servers** | Apache Tomcat, Nginx, Gunicorn, Node.js |
| **Monitoring & Logging** | Prometheus, Grafana, Kubernetes Metrics |
| **Cloud Technology** | AWS (ECS, EKS, Lambda), GCP (Cloud Run, Pub/Sub, BigQuery), Azure DevOps Pipelines, Docker, Kubernetes, Terraform |
| **Databases** | MongoDB, Oracle 19c (PL/SQL), PostgreSQL, MS SQL Server, Redis |
| **Version Control & Tools** | Git, GitHub, Jenkins, Cloud Build, GitHub Actions, Terraform, Jira |

**Professional Experience**

**Graduate Research Assistant (Data) – University of Nebraska–Lincoln** | ***Jan 2025 – Present***

* Built event‑driven ingestion pipeline on **GCP**: Canvas click events → **Pub/Sub → Flask** services on **Cloud Run → BigQuery**; back‑filled curated metrics to **Oracle** via **PL/SQL**.
* Developed real‑time **D3.js** dashboards; enforced **85% CI** coverage gate with **Cloud Build** and **pytest**.

***Environment****:* Python, Flask, D3.js, GCP Cloud Run, Pub/Sub, BigQuery, Oracle, Cloud Build, GitHub Actions

**Graduate Research Assistant (GIS) – University of Nebraska–Lincoln** | ***Aug 2024 – Dec 2024***

* Created **RESTful Flask API** serving **100 K+ geoJSON** **records**; cached hot tiles in **MongoDB + Redis**, reducing map load times **60%**.
* Deployed via **Docker** on **Linux VM** (**Nginx/Gunicorn**) with **Prometheus‑Grafana monitoring**.

***Environment****:* Python, Flask, MongoDB, Redis, Docker, Nginx, Prometheus, Grafana, Git

**Graduate Research Assistant (AI/ML) – University of Nebraska–Lincoln** | ***Nov 2021 – Dec 2023***

* Optimized scalable decision‑making algorithms (PBVI, I‑POMCP‑PF) by porting core loops to **Cython** and **CUDA**; runtime reduced **40%**.
* Built **pytest + Cython test harness** with continuous benchmarks (**GitHub Actions**); authored **Sphinx docs for C‑extension API**.
* Designed **R Shiny** dashboard visualizing **1 M+ policy trajectories**, cutting analysis time from days to minutes.

***Environment:***C, C++, Cython, Python, TensorFlow, GNU Debugger, Valgrind, R Shiny, Git

**Programmer Analyst – Cognizant, India** | ***Feb 2018 – Aug 2021***

* Delivered **12 Java 11 Spring Boot microservices** on **AWS ECS/EKS**, processing **3 M REST calls/day** with **p95 latency < 100 ms**.
* Implemented **MongoDB + Oracle dual‑store strategy**; wrote **PL/SQL** for legacy reporting, persisted high‑velocity events to **MongoDB Atlas**, cutting query latency **40%**.
* Refactored **250 K‑LOC J2EE** **monolith** into **micro‑fronted services**; introduced **Resilience4j** circuit breakers and **Eureka discovery**, achieving **99.95% uptime**.
* Led code‑review guild of **6** engineers; instituted **JUnit 5 + Mockito TDD workflow**, raising coverage **from 55% to 90%** and eliminating post‑release defects for two consecutive quarters.
* Drove **Agile Scrum** ceremonies across US–India team; **delivered 30% velocity lift**, mentored 5 new hires, and managed **blue/green Kubernetes upgrades** with **Prometheus‑driven SLOs**.

***Environment:***Java 11, Spring Boot 3, Spring Cloud Netflix, Kafka, MongoDB, Oracle PL/SQL, AWS EKS, Docker, Kubernetes, Jenkins, Git

**Selected Projects**

* **Personal Portfolio:** Architected and developed a responsive portfolio application with a React + TypeScript frontend (Vite, Tailwind CSS, PostCSS) featuring 3D robot animations powered by Three.js (OrbitControls, GLTFLoader), alongside a Spring Boot (Java 17) backend exposing secure REST APIs with Spring Data JPA, Hibernate, Flyway database migrations, and JWT-based authentication. Containerized both services using Docker multi-stage builds (Node→Nginx for UI, Maven→JRE for API) and orchestrated them via Docker Compose (with PostgreSQL in Docker and optional in-memory H2 for dev), ensuring production parity and streamlined local development.
* Live Website: [Bala SE Portfolio Website](https://polite-plant-0ce0f1c0f.6.azurestaticapps.net/)
* Please also find other projects showcased here.
* **Mini Equipment Marketplace**: Developed a full-stack equipment listing platform using ASP.NET Core MVC, Entity Framework Core, and SQL Server, featuring role-based access (Admin, Vendor, Shopper), dynamic UI with AJAX filters and modals, and secure authentication via ASP.NET Identity. Deployed to Azure App Service with GitHub CI/CD, integrated Azure SQL and environment-specific configurations, and implemented production-ready email workflows (registration, password reset, quote requests) using MailKit and SMTP.
* Live Website: [Equipment Marketplace Website](https://miniequipmarketplace-factgzd7cpeabne8.canadacentral-01.azurewebsites.net/)
* **Multi-Agent War Simulation:** Built a browser-based visualization of multi-agent reinforcement learning with a Flask backend (app.py, WebSimulationVisualizer, Matplotlib’s JSHTML animations) and a Bootstrap/vanilla-JS frontend supporting dynamic simulation settings (agent count, health, animosity, learning rate, discount factor, agent types). Containerized with Docker (Python 3.9-slim base, CPU-only PyTorch wheels, runtime-only image), hosted on AWS using a **public ECR** repository and **App Runner** (1 vCPU/2 GB, HTTP health checks on port 8000), secured via an IAM user with scoped policies and monitored with CloudWatch logs and default scale-down alarms.

**Education**

**M.S. in Computer Science – University of Nebraska–Lincoln | Aug 2021 – Aug 2025**

Key coursework: Advanced Software Engineering, Graph Algorithms, Cybersecurity & Cloud Computing, Databases, Multi‑Agent Systems

**B.Tech. in Electrical & Electronics Engineering – SRM Institute of Science & Technology | Aug 2013 – May 2017**