Balasubramanyam Evani

balasubramanyam.evani@gmail.com • (412) 726-8950 • Linkedin: balasubramanyam-evani • Github: BalasubramanyamEvani

EDUCATION

Carnegie Mellon University — Pittsburgh, PA
M.S in Electrical and Computer Engineering - specialization in Systems and Machine learning; **GPA: 3.94/4.0** — December 2023
Selected Coursework: Distributed Systems, Cloud Computing, Computer Systems, Deep Learning, Natural language Processing

Manipal University — Jaipur, India
B. Tech in Electronics and Communication Engineering; **GPA: 9.78/10.0** — June 2019

SKILLS

Languages: Python, Golang, Java, C/ C++
Platforms: Linux, Windows, GCP (Vertex AI, GKE, Agent Engine, Cloud Run, Cloud Tasks, AlloyDB, Secret Manager, Certificate Manager, Buckets), AWS (EC2, EMR)
Data Engineering: Apache Spark, Kafka, Samza, HDInsight
DevOps and Infrastructure: Docker, Kubernetes, Helm charts, Terraform
Software Development Frameworks: FastAPI, Flask, PyTest, Asyncio, Spring Boot, Spring Data, Junit
Databases Weaviate, Elasticsearch, Postgresql, MongoDB, Neo4j
Version Control: Git, SVN, Jenkins
Agent Frameworks: LangGraph, Google ADK, LangChain
ML-Frameworks: PyTorch, PySpark, TensorFlow, Keras, OpenCV, MMSegmentation, NumPy, SciPy, Scikit-learn, Pandas, SentenceTransformers

EXPERIENCE

R&D Software Engineer (Openstream Inc. | Bridgewater, NJ) — **May 2024 –Present**
Insurance Underwriting Assistant

- Built a **multi-agent system** using **Google ADK** to simulate **insurance underwriting workflows** for one of the largest insurance providers, integrating **deterministic components** where required for compliance. Agents included modules for **loss analysis, anomaly detection, and risk evaluation** leveraging both internal and web data sources.
- Implemented storage of agent responses to **GCP BigQuery** and artifact outputs to **GCP Cloud Storage**, enabling persistent, queryable audit trails and reusable intermediate results.
- Architected the application following **Python namespace conventions**, with each agent capability modularized into individual packages for **cleaner dependency management** and **faster deployment cycles**.
- Improved backend performance by making the entire framework **async-compatible**, reducing average execution time per capability from **15 minutes to 7 minutes** in high-throughput test runs.
- Designed and executed **performance tests** using **Locust** on **uvicorn-based REST** services to simulate **high-concurrency workloads (up to 100 concurrent users)**, and integrated performance testing into the **CI/CD pipeline** for every release.
- Wrote unit tests using **pytest**, **pytest-cov**, and **pytest-asyncio**, achieving **82% test coverage** across all capabilities delivered to the client.
- Maintained **code quality and consistency** using **black**, **pylint**, and **isort**, and integrated a **Makefile** to automate the setup of the **local development environment** and **lint/test steps**.

Infrastructure

- **Spearheaded the infrastructure setup** from scratch, creating reusable **Terraform modules** to provision key **GCP resources** including **Cloud Storage Buckets, GKE clusters with autoscaling, Cloud Run services,** and secret setup using **Secret Manager**.
- Designed and implemented **Helm charts** and **Terraform-based deployments** for **Neo4j**, **Elasticsearch**, and proprietary applications on **GKE**, integrating **Kubernetes Gateway Service** to provision a **regional internal load balancer**. Enabled secure access to GCP services (AlloyDB, Cloud Storage) using **Workload Identity Federation**, improving compliance and eliminating static credentials.
- Implemented **namespace-level segregation** in GKE to logically isolate components for application services, databases, and Dynatrace monitoring, improving observability, security, and deployment governance.
- Defined **Horizontal Pod Autoscaler (HPA)** manifests for in-house applications based on **CPU** and **memory utilization** thresholds, ensuring cost-effective scaling and optimal performance under variable workloads.
- Configured **dynamic persistent disk provisioning with WaitForFirstConsumer** volume binding mode in the **StorageClass** to ensure optimal disk allocation and avoid unnecessary resource fragmentation.
- Created a GCP **Cloud Tasks queue** module to enable asynchronous workload processing by a separate Cloud Run service.

Client Engagement & Support

- Participated in on-site workshops at client locations to present system design architecture, demonstrate workflows, and align implementation details with client expectations using detailed slide decks and live demos.
- Served as the primary point of contact during production deployments, including after-hours on-call duty to ensure smooth rollout and rapid rollback procedures if required. Also, acted as the first responder for incident resolution in production environment.

Intern - Software Engineer AI & Algorithm (Carl Zeiss Meditec | Dublin, CA) — **July 2023 – August 2023**
Worked with Artificial Intelligence & Decision Support team to investigate AI models for ophthalmic images

- Developed a modular framework to analyze AIRNet, SuperPoint, and SuperGlue models for retinal image registration. Conducted ablation studies comparing traditional OpenCV SIFT with Brute Force matching and RANSAC, and SuperGlue with Brute Force matching and RANSAC.
- Achieved a 90.71% improvement in MSE over traditional SIFT and RANSAC methods.
- Managed and scheduled deep learning model training using Singularity, Docker, and SLURM workload manager.

Senior Software Engineer (LTIMindtree Ltd. | Bengaluru, India) **June 2019 – January 2022**

- Built a centralized platform for regression test result visualization using **Angular 8**, **Angular Material**, **Spring Boot** (**Java 8**), and Netflix Eureka. Developed custom runtime plugins for Cucumber, JUnit, and TestNG, significantly improving test failure traceability and reducing decision turnaround time by replacing manual Excel-based tracking.
- Designed and implemented robust data models using **Hibernate** and **JPA**, enabling efficient and scalable persistence layer operations.
- Secured application access by implementing **Spring Security + JWT** for **authentication** and **role-based authorization**.
- Wrote comprehensive unit tests using **JUnit** and **Mockito**, ensuring high test coverage and maintainable code.
- Developed a **flowchart-driven automation tool** for **cloud infrastructure provisioning** on AWS, integrating services like Elastic Load Balancer, Auto Scaling Groups, and dynamic scaling policies. Used **Terraform** for streamlined and repeatable infrastructure orchestration.

ACADEMIC PROJECTS (CMU | Pittsburgh, PA) **January 2022 – December 2023**

Ride Sharing App Service

- Developed a Java-based Kafka Producer to stream real-time data from trace files, partitioned by blockId, and efficiently delivered messages to remote Kafka topics on an AWS hosted Samza cluster.
- Designed and implemented a Samza application to consume these streams and compute the highest-scoring driver-client pair within each block.

Iterative Processing with Spark

- Engineered a Scala-based Spark application to calculate PageRank values in a Twitter Social Graph dataset. Monitored the application on Azure HDInsight Cluster, using YARN UI for bottleneck identification.
- Performed an in-depth analysis and comparison of RDD and DataFrame APIs to derive meaningful insights from a large-scale Twitter dataset.

CMUD Backend

- Implemented an Actor Model in Golang to efficiently coordinate tasks across multiple machines, leveraging message passing for optimized communication. Ensured robust consistency guarantees for replicated data using streamlined RPC communications.
