Anthony Nikhil Reddy Lingala

+1 (347)-875-2294 al8291@nyu.edu linkedin.com/in/anthonynikhilreddy anthonynikhil.work

Education

New York University **Sep 2023 – May 2025**
Masters of Science in Computer Science. CGPA: 3.89 *New York, NY*

Vellore Institute Of Technology **Jul 2017 – Jun 2021**
Bachelor of Technology in Computer Science. CGPA: 3.34 *Vellore, India*

Certifications

AWS Certified Developer - Associate **Jul 2024**

Technical Skills

Programming Languages: Python, TypeScript, Java, SQL, JavaScript, C++, C
Frontend: React, Angular
Backend: Flask, Django, Node.js, Spring Boot, AWS Lambda, API Gateway
Cloud: AWS (DynamoDB, SQS, S3, EC2, ECS, ECR, CodePipeline), GCP (BigQuery, Cloud Functions, Cloud Run)
Devops: Docker, Jenkins, Grafana, Prometheus
Databases: PostgreSQL, MySQL, MongoDB, Redis, Neo4j
Data Engineering: Apache Spark, PySpark, Kafka, Hadoop

Experience

TagNTrac.ai **Jul 2025 - Present**
Software Development Engineer *Remote*
- Built and deployed backend API services integrating BigQuery NLP pipelines into analytics dashboards, improving data availability and cross-team insight delivery.
- Translated prototype ML/NLP models into fault-tolerant production APIs, ensuring reliable, controlled, and auditable data flows.

Tavant Technologies **Jun 2024 – Aug 2024**
Software Development Intern *Santa Clara, CA*
- Designed and implemented a Neo4j-based recommender system and integrated it with Python data pipelines, improving recommendation accuracy by 30% while automating ingestion and preprocessing workflows.
- Built auxiliary data processing services using AWS Lambda and S3, increasing pipeline efficiency by 35% and reducing manual intervention by 40%.

Verizon **Aug 2021 - Jul 2023**
Software Development Engineer - 2 *Hyderabad, India*
- Designed and maintained distributed data ingestion pipelines on AWS EC2, SQS, SNS using Python, improving data accuracy by 25% and reducing manual intervention by 70%.
- Implemented serverless workflows with AWS Lambda and Step Functions backed by DynamoDB, lowering operational cost by 35% while increasing workflow reliability.
- Automated ingestion from 300+ Jenkins jobs into PostgreSQL using PySpark and Hadoop, decreasing processing latency by 65%.
- Built a React-based engineering dashboard integrated with New Relic APIs and REST services, providing real-time visibility into system and portfolio health.
- Developed unified CI/CD pipelines and operational dashboards using Jenkins, Grafana, and Prometheus, improving deployment velocity and resource efficiency by 20%.
- Optimized high-throughput streaming pipelines using Apache Spark and Kafka, reducing end-to-end latency and improving system performance; served as DRI for diagnosing and restoring pipeline failures.

Projects

EasyTex.cc — AI-Powered LaTeX Resume Builder | *AWS, React, Node.js, Python, LLMs*
- Developed and launched EasyTex.cc, an AI-First resume builder used by 1,500+ users for generating ATS-friendly LaTeX resumes.
- Implemented LLM-based content generation for bullet rewriting, skill extraction, and resume optimization, reducing manual editing time for users.
- Designed a scalable multi-tenant backend on AWS with secure authentication, template rendering services, and low-latency PDF generation workflows.

Multitenant Database as Service | *AWS EC2, Redis, SQS, SSM, MySQL*
- Designed a multi-tenant DBaaS system capable of dynamically onboarding up to 50 tenants per second, reducing manual provisioning time by over 80%.
- Improved schema migration success rate to 95% through automated SQL validation, Redis-backed retry mechanisms, and asynchronous workflows.
- Reduced database creation latency by 60% compared to RDS using MySQL on EC2, achieving sub-0.01-second initialization times.

Collaborative Online Code Editor | *AWS, Y.js*
- Architected a high-performance serverless collaborative code editor using AWS Lambda, WebSockets, Y.js, API Gateway, and DynamoDB, achieving 99.9% uptime while supporting 100+ concurrent users.
- Engineered a distributed code execution platform using Docker containers orchestrated through AWS Fargate and Kubernetes, implementing language-specific sandboxes with Flask/Python that processed 1,000+ concurrent code executions per second
- Established a robust CI/CD pipeline using AWS CodePipeline, GitHub Actions that reduced deployment time by 60% and eliminated downtime.
- Optimized system reliability and fault tolerance, leveraging AWS Auto Scaling, Elastic Load Balancing, and Redis for distributed caching, dynamically provisioning resources to handle high-concurrency workloads.