BHASKAR REDDY VANTEDDU

+1 (510) 306-4868 | bhaskar07v@gmail.com | Portfolio | LinkedIn | GitHub | San Francisco, CA

EDUCATION

University of Central Missouri **AUG 2023 – MAY 2025**
Master's in Computer Science

Jawaharlal Nehru Technological University
Bachelor's in Computer Science & Engineering

TECHNICAL SKILLS

Certifications: Combined Excellence Certification - DataEngineer.io (led by Zach Wilson)

Programming & Development: Python | Boto3 | AWS Services | Unix | Pandas | Docker | Flask | Git | CI/CD

Data Frameworks: PostgreSQL | PySpark/Spark | Snowflake | Iceberg | Trino | Apache Airflow | DBT | Databricks | Kafka | Tableau|

Cloud Frameworks: AWS Glue | AWS S3 | AWS RDS | AWS DynamoDB | AWS Redshift | AWS EMR | AWS Lambda | Astronomer

AI Frameworks: Agentic AI systems | MCP(Model Context Protocol) | LLM | Predictive Analytics | Prompt Engineering

WORK EXPERIENCE

N Folks Solutions (Data Engineering Intern) **JAN 2023 - MAR 2023**

- Designed a scalable, real-time event processing pipeline ingesting ~1M website events/hour from Confluent Kafka to Apache Iceberg using PySpark on Databricks, implementing offset-limiting (10k records/batch) for fault tolerance.
- Engineered a medallion architecture with progressive transformations: binary-decoded bronze tables, schema-validated silver tables with nested structure normalization, and enriched gold tables with geographical data, ensuring data quality.
- Constructed advanced streaming analytics using tumbling windows (1-minute windows with 30-second watermarks) and session windows (5-minute inactivity threshold), ensuring 99% data completeness for accurate geographic traffic patterns while enabling analysis on infrastructure resources based on real-time user engagement metrics.
- Developed an IP-based location enrichment system with custom caching mechanism for ipinfo.io API calls, reducing average latency by 6 seconds during peak traffic periods, and consolidated an interactive Databricks dashboard that visualized regional engagement patterns, enabling efficient infrastructure resource analysis based on geographic trends.

Morse Team (Data Engineering Intern) **MAY 2022 - JUL 2022**

- Architected a space-efficient cumulative table in Spark SQL using binary array encoding to represent daily user events, achieving ~30x storage reduction by merging 30 daily rows into one monthly row while retaining all the information.
- Migrated dimension table architecture to track complete change history rather than just start/end states, enabling comprehensive historical data tracking and supporting more granular time-based analysis for improved business insights.
- Orchestrated Airflow DAGs with sequential execution to maintain data integrity during dimension table backfilling, ensuring accurate processing of time-dependent datasets where each day depends on previous results.
- Optimized Spark jobs through collaboration with analysts and communication with non-technical stakeholders in a cross-functional agile development environment to identify essential columns and preserve Parquet run-length encoding, reducing table size by 50% and decreasing backfill processing time by 60%, thereby improving big data ETL efficiency.

PROJECTS

Hyper-ADS: An Agentic AI Ad Assistant

- Introduced an AI-driven ad recommender merging real-time event data and weather forecasts, ensuring small shops never miss prime sales windows. Reduced decision-making time from 3+ hours to 5 minutes per day while handling 500+ events daily.
- Constructed a complete ETL pipeline that lifts cleaned event data into RDS and merges live weather details. Empowered a suite of AI agents to self-generate daily ad strategies in a server on AWS App Runner, cutting human oversight to zero.
- Architected a cloud-native system using AWS Lambda for scheduled scraping, RDS PostgreSQL for storage, and Server-Sent Events for real-time updates. Employed Python's async capabilities alongside agentic AI with Gemini models to process complex decisions.
- Generated 2K+ marketing recommendations monthly, analyzing local events with weather data. Boosted client engagement by 65% through targeted timing suggestions while processing 12,000+ monthly events, saving businesses $400/month in ad spend.

Stocks Analysis

- Structured parallel ingestion of unstructured data from Polygon API using Spark's repartitioning strategy with 4 concurrent executors instead of sequential driver calls, achieving 4x faster ingestion, supporting over 10k calls per minute, and lowering AWS Glue costs.
- Implemented a list-based temporal schema in Apache Iceberg (S3-backed) with date-based partitioning for stock metrics (open/close, highs/lows, volumes, pre/after-hours), enabling efficient storage and historical analysis through dynamic unnesting.
- Integrated data quality framework with Pytest and Chispa for deduplication checks, null-value detection, and noise filtering. Also validated core logic with unit and integration tests following data governance standards.
- Automated incremental data processing with Apache Airflow DAGs for reliable workflow execution, ensuring data completeness while minimizing processing overhead through parameterized task dependencies.

Actors Historical Analysis

- Analyzed Hollywood actors' film history (1914-2021) using a year-list data structure that consolidated annual metrics into single array elements, **leading to a >25% decrease in storage footprint**, facilitating rapid historical analytics.
- Applied strategic data modeling to construct a dimension table in Trino with Iceberg, adopting array-based cumulative methods to capture full actor change histories, enabling thorough performance tracking and point-in-time analysis.
- Established quality with DBT tests (not null, unique, accepted values) and WAP pattern following industry best practices, catching anomalies before production and reducing numerous post-deployment fixes.
- Built idempotent, incremental pipeline with Airflow Scheduler that automatically ingests latest data and merges with master history table, using proper DAG dependencies and deduplication checks for near-zero duplications during backfill.