Nischith Javagal Panish
nischithpjavagal@gmail.com
+1-(817)-420-0401
Portfolio - https://nischithp.github.io/
LinkedIn - https://www.linkedin.com/in/nischith-javagal/
GitHub - https://github.com/nischithp

Education

Master of Science in Computer Science (MS) - The University of Texas at Arlington - 2021
Bachelor of Engineering - Computer Science(B.Tech/B.E) - Vishweshwaraya Technological University - 2019

Summary

Data Engineer and Software Developer with 3+ years of experience in building scalable, high-throughput data pipelines and real-time ETL solutions using AWS (ECS, Lambda, Kinesis, RDS) and Kafka. Skilled in Python, Kotlin, and Java, with expertise in cloud-native architectures and Snowflake integration for robust analytics. Passionate about optimising data workflows to enhance performance, reduce costs, and deliver reliable insights.

Skills

Languages: Python, Kotlin, Java
Cloud Technologies : AWS -, Elastic Container Service(ECS), Elastic Container Registry(ECR), Redis, DynamoDB, Lambda Functions, CDK, Aurora(RDS), Kinesis, SQS, IAM, CodePipelines
Other Technologies: Kafka, High Throughput Real Time ETL/Data Pipelines
Database Technologies: MySQL, Snowflake, PostGreSQL
Additional Skills: Git, Microservices, API Development, Full Stack, Mobile/Web Application Development

Work Experience

Global Traffic Technologies/Miovision – July 2021 – Present, Austin, Texas - Full time

- Led a cross-functional initiative to modernize data ingestion and ETL architecture, decommissioning legacy pipelines and deploying robust, real-time data pipelines tailored for analytics.
- Designed and implemented high-throughput data pipelines using Apache Kafka, with the service producing ~30 million records per hour and handling ~900 million records daily.
- Architected and implemented a scalable data ingestion and IoT message processing system capable of handling 500 messages per second.
- Introduced a caching mechanism, slashing response times by over 95% from 1100ms to 50ms using Redis.
- Simultaneously led the development of EVP and TSP solutions, enhancing traffic management and reducing response times for emergency vehicles and public transit in major metropolitan areas across North America.
- Recognized the need for a centralised Feature Toggle Service and spearheaded its development, enabling seamless feature management across teams and eliminating the need for redundant application stacks or hardcoding feature flags, enabling dynamic feature management and reducing development time for agencies.
- Worked with the Deployment team and wrote cloud-formation/Terraform templates (IaC).
- Streamlined existing project and improved product performance. Implemented data validation checks at front end and back end layers using Pydantic.

Projects

Transit Signal Priority (Skills Used: Spring, ECS, Lambda Functions, CDK, Redis, ECR) - GTT/Miovision

- Responsible for the development and architecture of the Transit Signal Priority (TSP) system, encompassing device support, data validation, real-time data handling, and cloud infrastructure using AWS Fargate, ECS, and CDK, all orchestrated through an event-driven architecture powered by Redis Pub/Sub.
- Implemented device support for IoT-based hardware devices within the TSP system, enabling seamless communication and data exchange, leveraging Python as the primary development language.
- Designed and developed data validation mechanisms using Pydantic to safeguard the integrity of incoming realtime data streams within the TSP system, ensuring data reliability and consistency, while also constructing a data aggregation service to support playback and archival of live requests sent to intersections, all accomplished using Python and hosted on AWS.

Emergency Vehicle Preemption (Lambda Functions, IoTCore, DynamoDB, Redis, Cloudformation) - GTT/Miovision

- Successfully migrated three distinct device families from Neptune to DynamoDB, streamlining data management and improving system performance.
- Implemented a Redis cache and cache invalidation mechanism utilising SQS, SNS, and AWS Lambda, optimising data retrieval and reducing latency for all devices.

API based Video Sharing Platform on Google Cloud – (GitHub) (Cloud Functions, REST API, JSON, Python, Flask, Cloud Storage, HTML, CSS, Bootstrap, Kubernetes)

- Architected and deployed RESTful APIs on Cloud Functions, integrating with Cloud MySQL and GCS.
- Developed a Python-Flask website hosted on GKE, enabling video upload and viewing, and managed container images, YAML files, and Kubernetes cluster deployments for public exposure.