Case Study:

E-Commerce Product Catalog Ingestion

We assisted a mid-sized fashion marketplace company to streamline and scale the ingestion of product catalogs from various retailers. This catalogs data is extremely important for company to generate personalized clothing recommendations for buyers.

Company overview

Client name: Confidential

Services: Marketplace, Personalized fashion

Technology: GCP Composer, Kafka, AirByte

Industry: E-Commerce

Location: US

- The client is a consumer-facing fashion marketplace offering a personalized shopping experience.
- It provides a singular destination where shoppers can explore thousands of leading clothing brands.
- The marketplace aims to curate products to each shopper's personal style, brands, fit, and size.

The challenges

- ➔ The client faced challenges in scaling catalog ingestion process due to growing number of retailers and increasing catalog sizes.
- ➔ Keeping catalog data up-to-date for good user shopping experience.
- ➔ Provide raw catalog and user demographics data to the Data Science team to generate personalized product recommendations.
- ➔ Minimize latency and optimize resource utilization while consuming millions of catalog products.
- ➔ Ensure fault-tolerance and data consistency in end to end processing.
- ➔ Simplify the onboarding of new retailers to ease integration with marketplace.

Case Study:

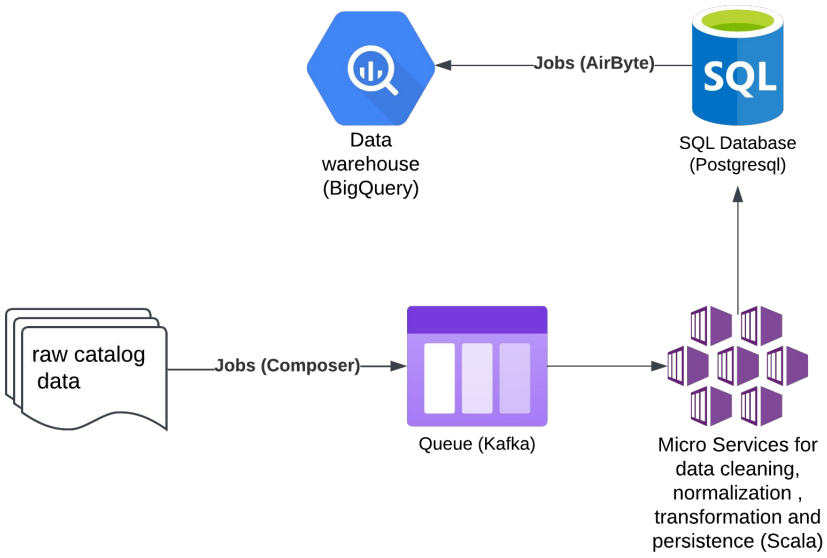
E-Commerce Product Catalog Ingestion

Outcome

The solution delivered multiple benefits through the use of various GCP services and Kafka.

- **Real time data processing**: Kafka enabled real-time streaming of data, which is particularly useful for catalog updates (e.g., price changes, inventory levels). This ensures that catalog reflects the most current information.
- **Scalability:** Handle large volumes of data, especially during peak loads (e.g., seasonal e-commerce updates). Scale microservices horizontally to accommodate growth in data sources and catalog size.
- **High throughput:** Scala's efficiency ensures that services can handle large-scale catalog ingestion workloads without significant performance degradation.
- **Idempotent Processing**: Ensure reprocessing the same data does not cause duplication or inconsistency which saves cost.
- **Data Lineage and Auditing**: Track the origin and history of data for troubleshooting and compliance purposes.

The approach



The solution

- The architecture leveraged GCP's managed services, along with open-source tools and Kafka for data ingestion and orchestration.
- We chose Kafka to stream catalog data because it is designed to handle large volumes of data efficiently.
- **Kafka** served as the backbone for real-time data streaming. A **GCP Composer** batch job pushed raw catalog data to Kafka topics, ensuring **decoupling** between ingestion and processing layers.
- **Microservices** were created using **Scala** and Kafka streams scala client libraries.
- Real-time transformations on the kafka topics were then applied and transformed catalog data was being persisted in **Postgresql** DB.
- **Airbyte** was used to ingest data from various sources (Postgresql, ftp, etc.) to **GCP BigQuery** to be consumed by the data science team to generate product recommendations.

For more details, please send your inquiry to info@pizenith.com
or visit our website www.pizenith.com