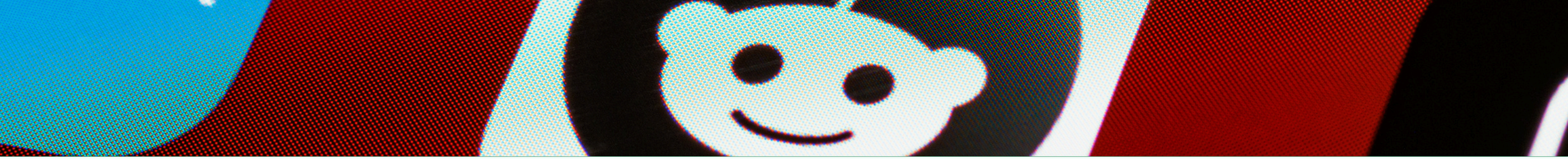
Scalable ETL Pipeline on Google Cloud

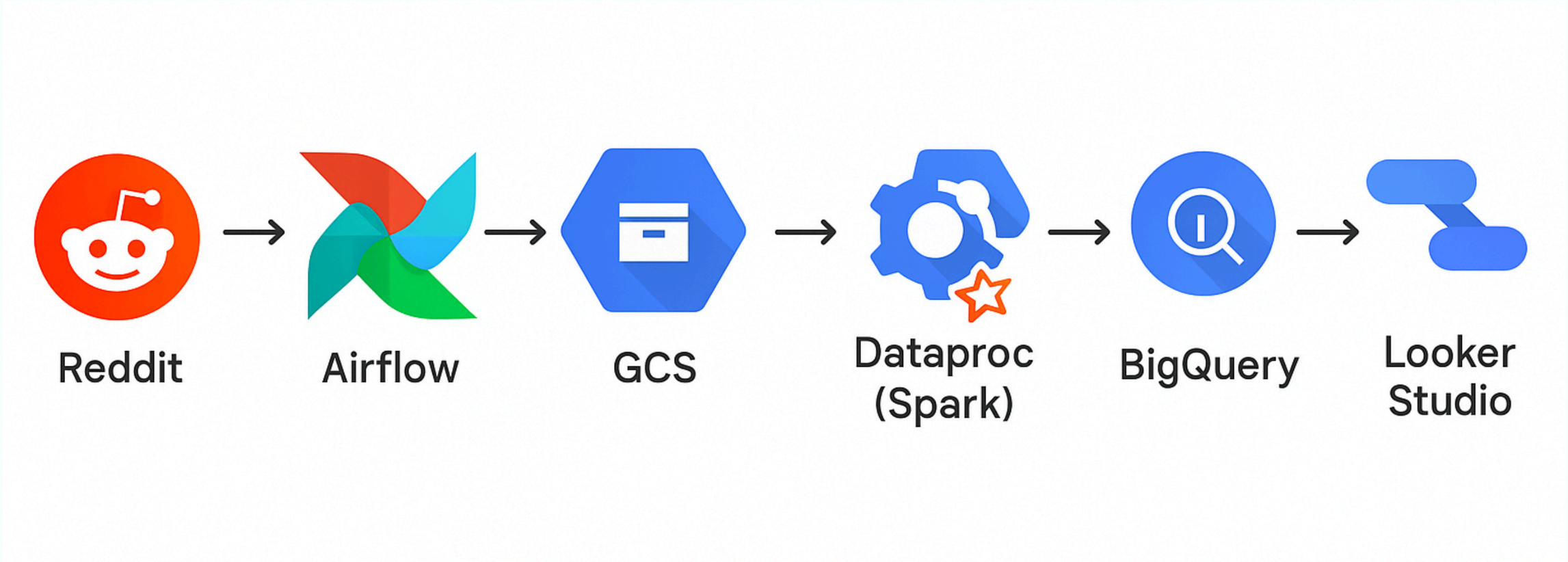
REDDIT STOCK SENTIMENT



Introduction & Motivation

- Social media (Reddit) influences stock market sentiment.
- Analyzing Reddit data helps understand trends and public opinion.
- Goal: Build an automated pipeline to collect, process, and visualize Reddit stock discussions.

High-Level Architecture Diagram



Data Pipeline Overview

Step 1: Ingest Reddit data (Python script)
Step 2: Convert JSON → CSV → Parquet
Step 3: Upload to GCS
Step 4: Create external and partitioned tables in BigQuery
Step 5: Run Spark job for word count analysis
Step 6: Visualize results in Looker Studio

Detailed Pipeline Flow

- **Airflow DAG:** manages all steps and dependencies.
- **Python Operator:** Runs custom scripts for data extraction and transformation.
- **Bash Operator:** Cleans up local files.
- **BigQuery Insert Job Operator:** Creates and manages tables.
- **Data proc Submit Job Operator:** Submits Spark jobs for heavy processing.

Tools & Technologies Used

- **APACHE AIRFLOW:** WORKFLOW ORCHESTRATION
- **GOOGLE CLOUD STORAGE (GCS):** DATA STORAGE
- **GOOGLE DATAPROC (SPARK):** DISTRIBUTED DATA PROCESSING
- **GOOGLE BIGQUERY:** DATA WAREHOUSING AND ANALYTICS
- **LOOKER STUDIO:** DATA VISUALIZATION
- **PYTHON:** SCRIPTING AND ETL LOGIC

Data Ingestion

- **Source:** Reddit API (subreddits related to stocks)
- **Fields Collected:** id, title, author, num_comments, total_awards_received, submission_date
- **Storage:** Raw JSON files

Data Processing & Storage

- **Transformations:** JSON → CSV → Parquet (efficient, columnar storage)
- **Upload:** Parquet files to GCS for scalable storage and easy access by BigQuery

Data Warehousing in BigQuery

- **External Table:** Reads Parquet files directly from GCS
- **Partitioned Table:** Organizes data by date for efficient querying
- **Schema Example:** id, title, author, num_comments, total_awards_received, submission_date

Visualization with Looker Studio

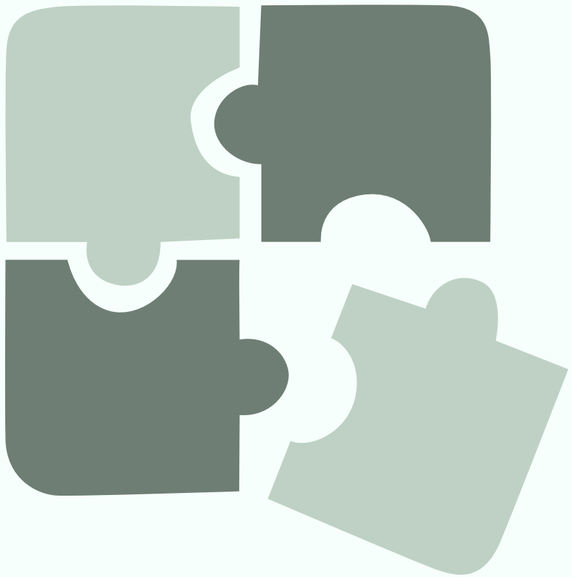
Challenges Faced

- **Cluster Management:** Creating new clusters was slow; switched to reusing existing clusters
- **Data Format Compatibility:** Needed multiple conversions for efficiency
- **Scheduling & Dependencies:** Ensured correct task order in Airflow
- **Cost Optimization:** Managed cloud resources to avoid unnecessary charges
- **Data Quality:** Handled missing or inconsistent Reddit data

---

Key Learnings

- **Orchestrating complex pipelines with Airflow**
- **Distributed processing with Spark**
- **Cloud data warehousing with BigQuery**
- **Building interactive dashboards with Looker Studio**
- **Importance of automation and monitoring**




thank
you