Building a Cricket Statistics Data Pipeline using Google Cloud

Team Members:

Likhith S (1RVU22CSE091)

Manvi Sinha (1RVU22CSE100)

Prince P Balar (1RVU22CSE124)

Syed Mohammed Wasiq (1RVU22CSE177)

Abstract

This project demonstrates how to build an end-to-end data engineering pipeline for cricket statistics, retrieving data from the Cricbuzz API, storing it in cloud storage, processing & ingesting it into a data warehouse, and finally visualising via a dashboard. The pipeline is built using Google Cloud services, demonstrating how raw sports data can be converted into meaningful analytics.

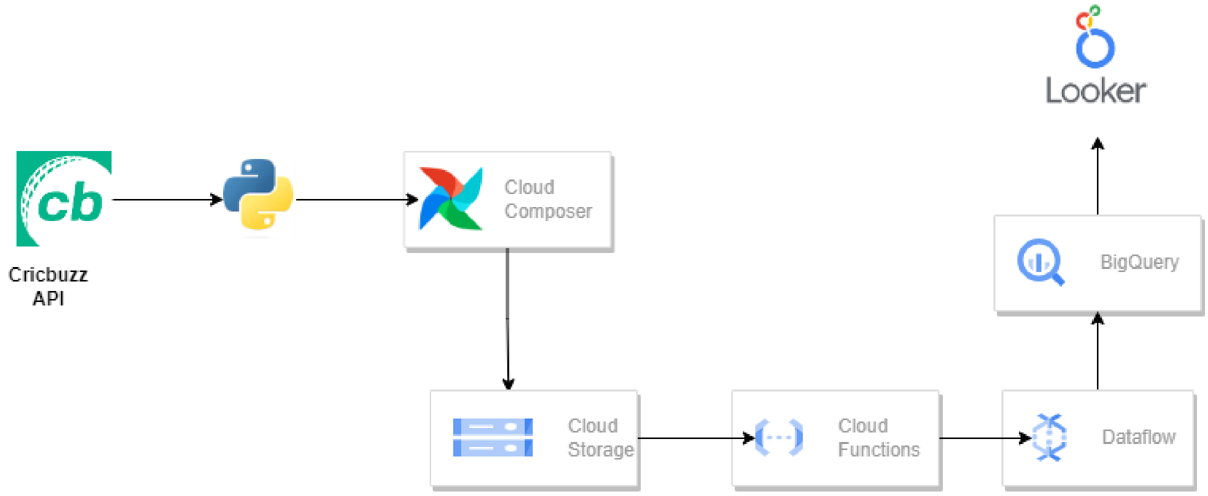
Introduction

- Background: Sports analytics is increasingly important, and cricket has rich statistics that lend themselves to data-driven insights.
- Problem statement: Manually gathering and analysing cricket match/player statistics is laborious. A robust pipeline automates the extraction, storage, processing and visualisation.
- Objective: To design and implement a reliable data engineering pipeline that:
  - Extracts cricket statistics via API
  - Stores raw data in cloud storage (Google Cloud Storage – GCS)
  - Processes the data and loads it into a data warehouse (BigQuery)
  - Builds a dashboard (Looker Studio) for visual analytics

System Architecture

- **Extraction:** Python scripts that call the Cricbuzz API and push CSV data to GCS.
- **Trigger:** A Cloud Function is configured to respond to file uploads in the GCS bucket.

- **Processing:** The Cloud Function invokes a Dataflow job (Apache Beam), which transforms the CSV and loads it into BigQuery.
- **Storage:** BigQuery acts as the data warehouse, storing structured tables of cricket stats.
- **Visualisation:** Looker Studio connects to BigQuery as the data source and presents dashboards (Looker.png) for analysts.



Technologies & Tools

- Python (main extraction/orchestration code)
- JavaScript (for any user-defined functions, e.g., udf.js)
- Google Cloud Platform (GCP) services:
    - Google Cloud Storage (GCS)
    - Cloud Functions
    - Dataflow (Apache Beam)
    - BigQuery
    - Looker Studio
- CSV, data ingestion & transformation pipelines
- API integration with Cricbuzz for cricket statistics
- DAG management (dag.py) to schedule/task orchestration

Methodology

1. **Data Retrieval**

    - The Python script extract_data.py calls the Cricbuzz API, fetches statistics (players, matches, rankings, etc.) and writes them into CSV format.
    - These CSVs are uploaded to a GCS bucket.

2. **Storage & Trigger Setup**

    - Once a CSV file is uploaded to the GCS bucket, a Cloud Function (in trigger_df_job.py) is triggered.
    - The function extracts metadata (file locations, parameters) and initiates a Dataflow job.

3. **Data Processing & Ingestion**

    - The Dataflow job (defined in dag.py or within extract_and_push_gcs.py) reads the raw CSVs, cleans/normalises data, applies transformations (including any UDFs in udf.js), and writes the resulting tables into BigQuery.
    - The project uses partitioning, schemas, and perhaps staging tables to manage data. (Details can be obtained by looking into the code.)

4. **Visualisation**

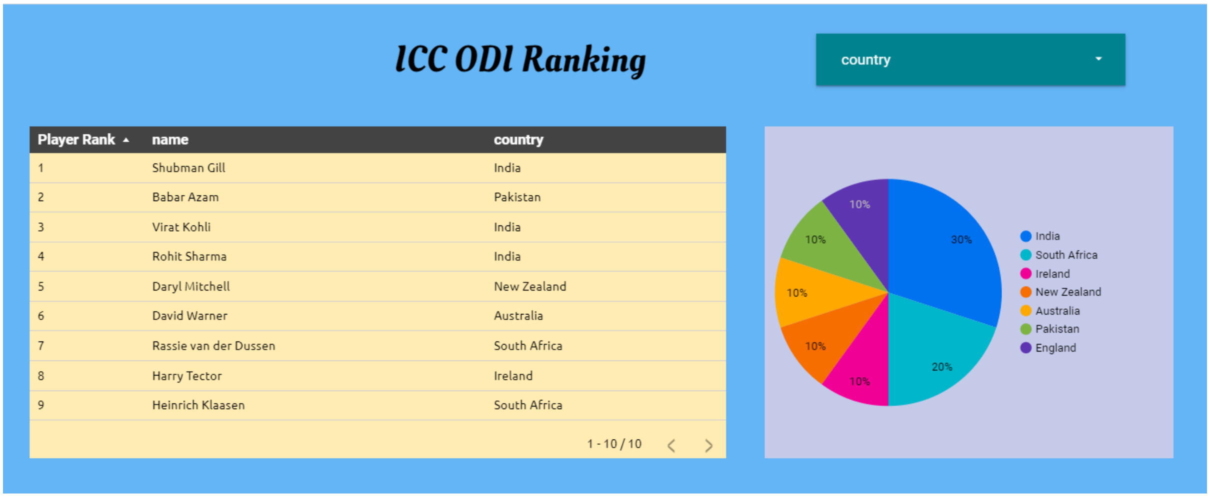
    - After data is loaded into BigQuery, Looker Studio is used to build dashboards: e.g., player rankings, batting/bowling trends, etc.
    - The Looker.png file in the repo shows the dashboard layout.

5. **Orchestration & Scheduling**

    - The DAG (Directed Acyclic Graph) defined in dag.py helps schedule and orchestrate the entire pipeline (e.g., retrieval → upload → trigger → processing → load → refresh).
    - The requirements.txt lists required Python packages for the pipeline.

Results

- The pipeline successfully loads cricket statistics into BigQuery.
- Sample outputs: The CSV file batsmen_rankings.csv is one example of ingested data.
- The dashboard displays interactive visualisations of batting rankings, match/player statistics.
- The pipeline is automated and scalable (leveraging cloud infrastructure) so that future data updates can be handled with minimal manual intervention.



ICC ODI Ranking

country

Player Rank ▲	name	country
1	Shubman Gill	India
2	Babar Azam	Pakistan
3	Virat Kohli	India
4	Rohit Sharma	India
5	Daryl Mitchell	New Zealand
6	David Warner	Australia
7	Rassie van der Dussen	South Africa
8	Harry Tector	Ireland
9	Heinrich Klaasen	South Africa

1 - 10 / 10

Conclusion

This project demonstrates a full-fledged data engineering pipeline for sports analytics. By leveraging modern cloud services on GCP, the pipeline extracts, processes, stores, and visualises cricket statistics efficiently. It offers a repeatable framework for similar data-intensive tasks in other domains.