Airbnb Big Data Analytics Project Documentation

1. Project Description

Overview

The project focuses on analyzing real-time Airbnb user actions using a big data analytics framework. Airbnb, a global online marketplace, connects travelers with hosts offering unique lodging experiences. The platform relies heavily on big data to optimize pricing, analyze user preferences, and provide secure and seamless transactions.

Significance

- Manages millions of listings worldwide, enabling real-time bookings.

Key Problem Solved

Efficiently matching hosts with guests by analyzing large-scale data on listings, user preferences, and market trends.

Role of Big Data

- Processes many bytes of structured and unstructured data.
- Provides real-time insights for enhanced decision-making.
- Supports scalability, fault tolerance, and low-latency data querying.

2. Design Challenges

Challenge 1: Scalability

- Managing high volumes of data generated by Airbnb's global operations.
- Ensuring the architecture scales horizontally to accommodate growing data.

Challenge 2: Real-Time Data Streaming

- Handling and processing real-time data from user interactions and IoT devices.
- Maintaining low latency during ingestion and processing.

Challenge 3: Data Consistency

- Ensuring consistent data across tools like Kafka, Spark Streaming, and Cassandra.
- Managing schema evolution and compatibility.

Challenge 4: Intuitive Visualization

- Designing an easy-to-navigate Streamlit dashboard.
- Providing actionable insights through interactive visualizations.

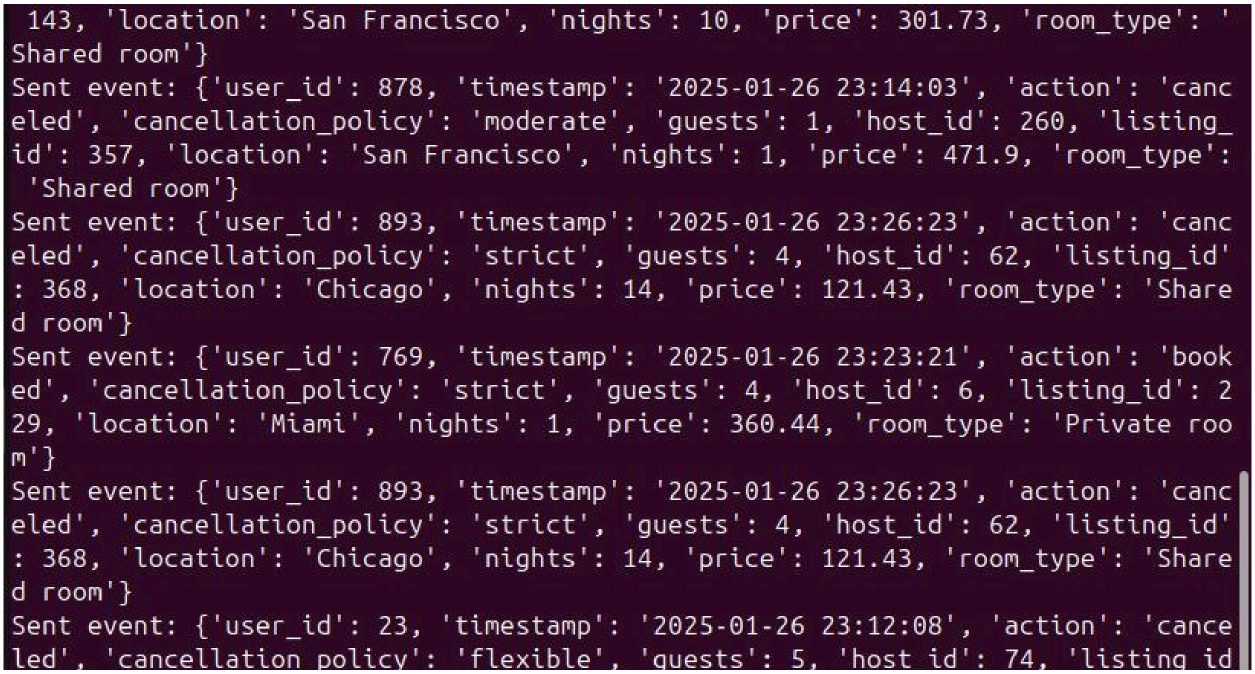
---

3. Big Data Architecture and Tools

Architecture Design

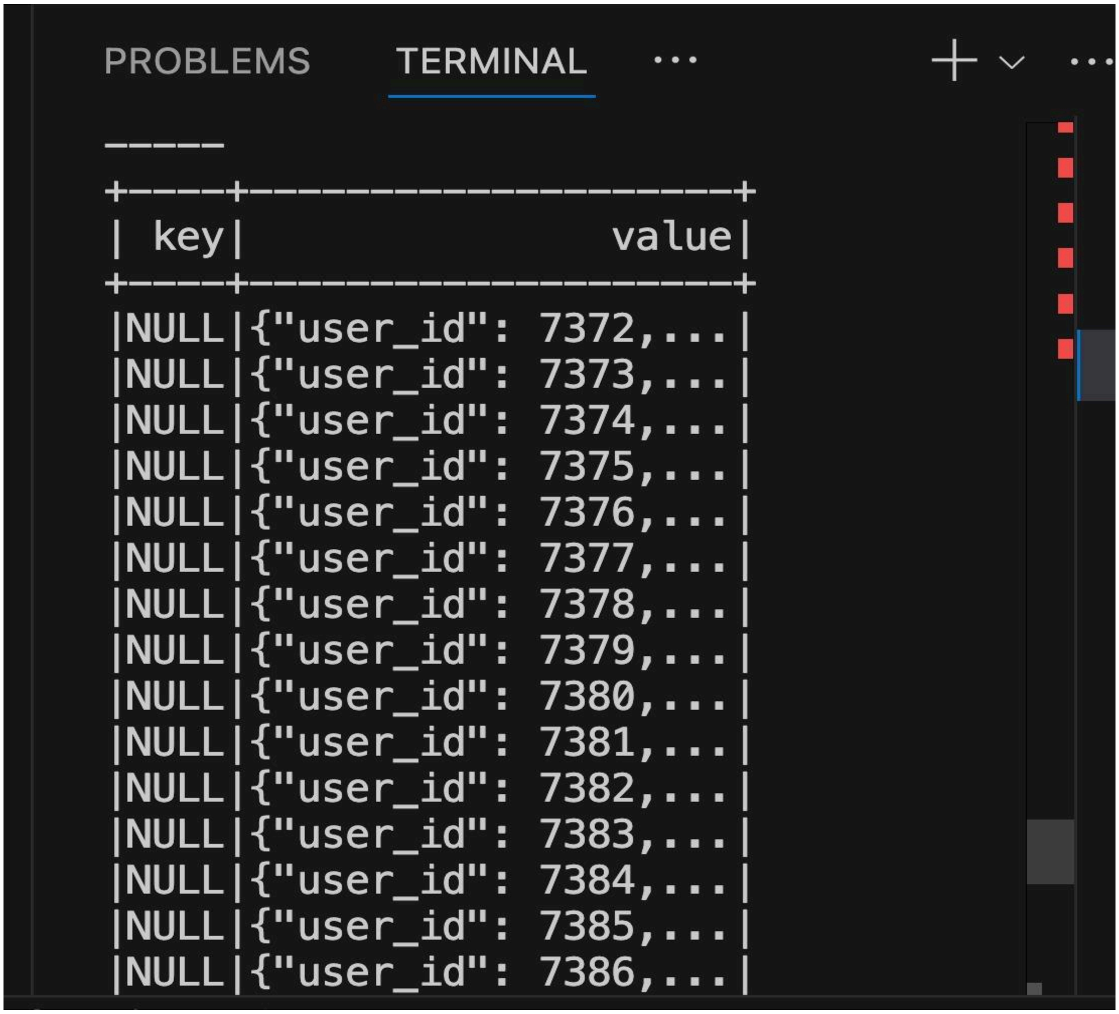
Data Flow Description

1. **Data Ingestion**:
   - User interactions and IoT device data are ingested via Kafka.
   - Third-party APIs provide additional data streams.



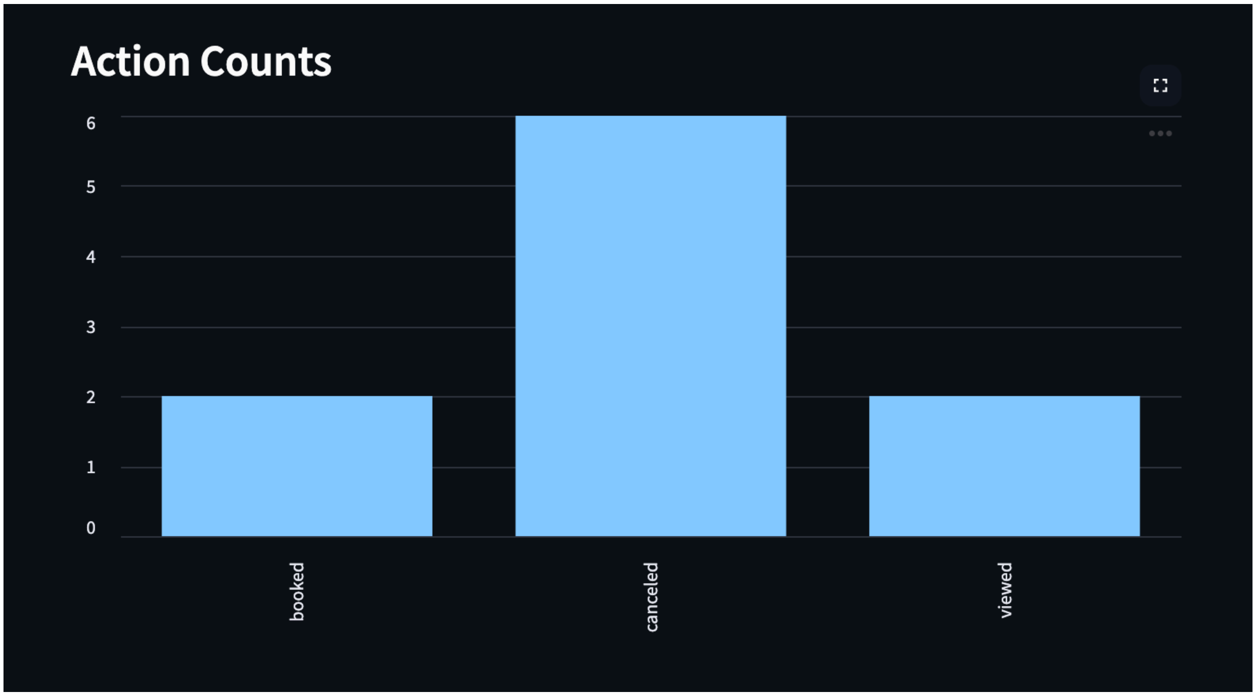
```
 143, 'location': 'San Francisco', 'nights': 10, 'price': 301.73, 'room_type': '
Shared room'}
Sent event: {'user_id': 878, 'timestamp': '2025-01-26 23:14:03', 'action': 'canc
eled', 'cancellation_policy': 'moderate', 'guests': 1, 'host_id': 260, 'listing_
id': 357, 'location': 'San Francisco', 'nights': 1, 'price': 471.9, 'room_type':
 'Shared room'}
Sent event: {'user_id': 893, 'timestamp': '2025-01-26 23:26:23', 'action': 'canc
eled', 'cancellation_policy': 'strict', 'guests': 4, 'host_id': 62, 'listing_id'
: 368, 'location': 'Chicago', 'nights': 14, 'price': 121.43, 'room_type': 'Share
d room'}
Sent event: {'user_id': 769, 'timestamp': '2025-01-26 23:23:21', 'action': 'book
ed', 'cancellation_policy': 'strict', 'guests': 4, 'host_id': 6, 'listing_id': 2
29, 'location': 'Miami', 'nights': 1, 'price': 360.44, 'room_type': 'Private roo
m'}
Sent event: {'user_id': 893, 'timestamp': '2025-01-26 23:26:23', 'action': 'canc
eled', 'cancellation_policy': 'strict', 'guests': 4, 'host_id': 62, 'listing_id'
: 368, 'location': 'Chicago', 'nights': 14, 'price': 121.43, 'room_type': 'Share
d room'}
Sent event: {'user_id': 23, 'timestamp': '2025-01-26 23:12:08', 'action': 'cance
led', 'cancellation_policy': 'flexible', 'guests': 5, 'host_id': 74, 'listing_id
```

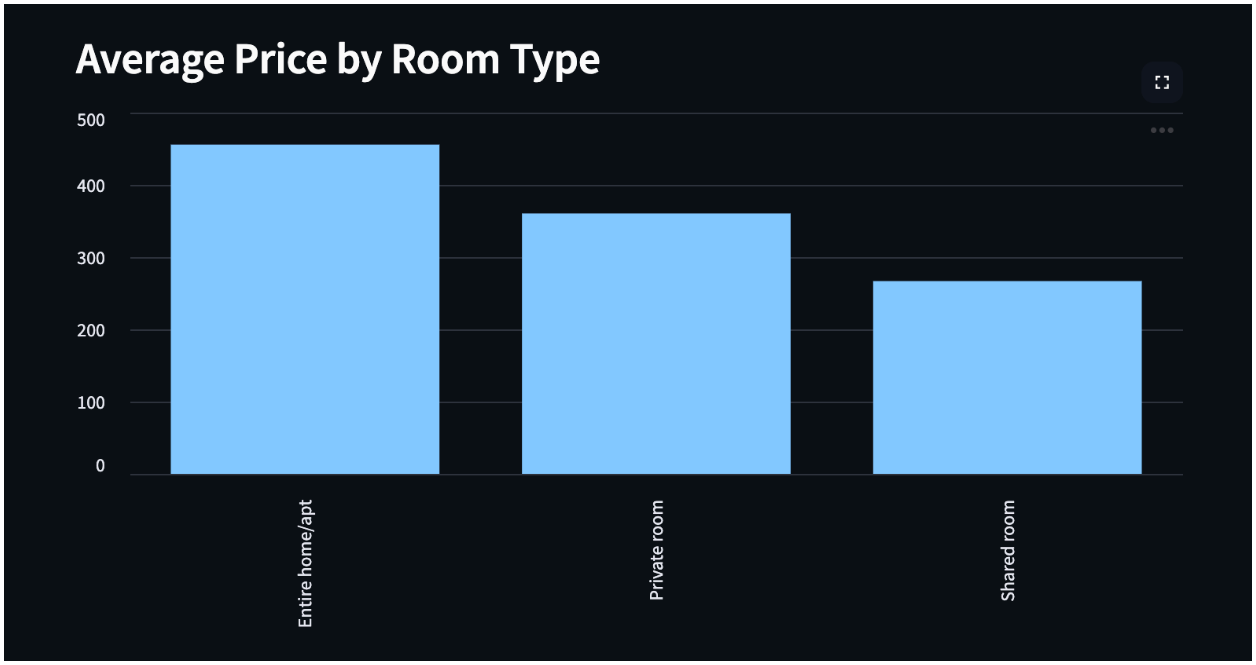
2. **Data Storage**:
    - **Cassandra** for real-time, low-latency data queries.
3. **Data Processing**:
    - **Spark Streaming** processes and enriches real-time data streams.

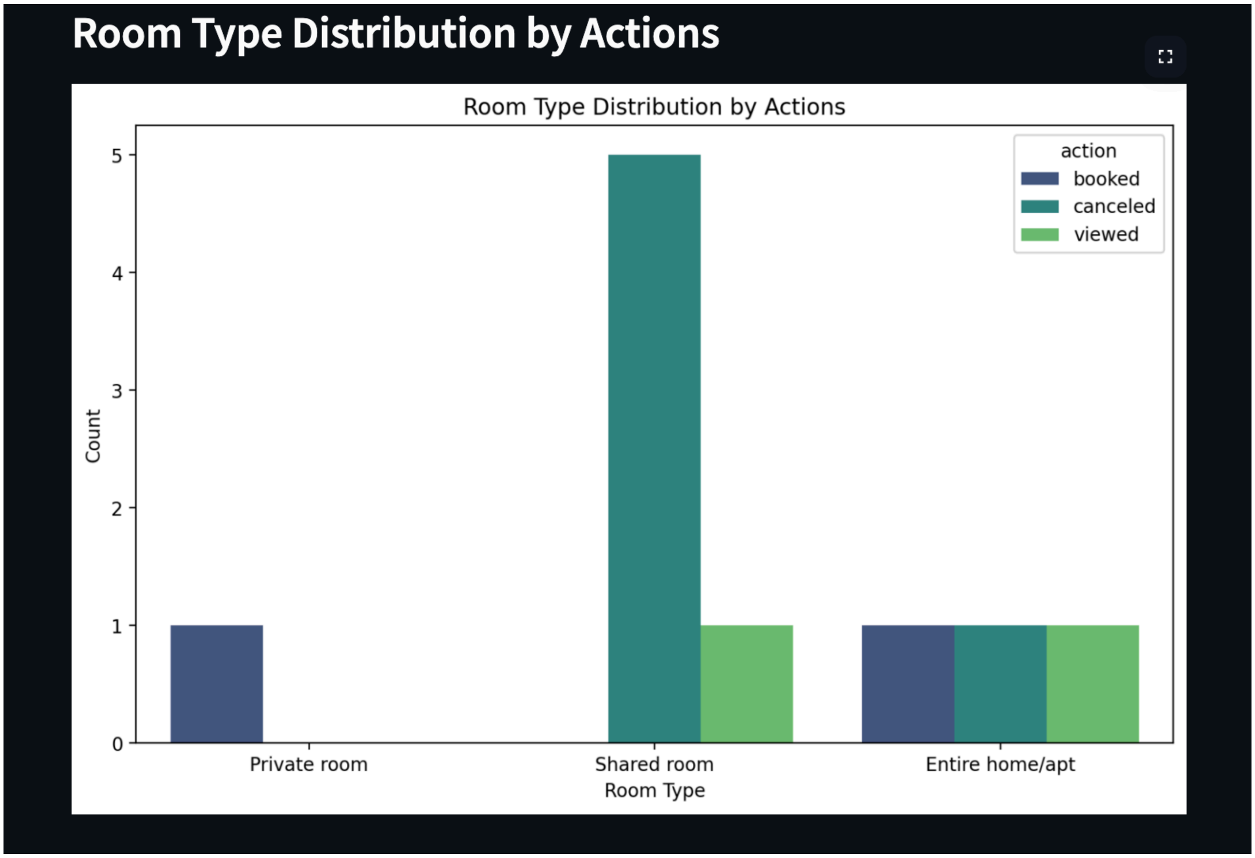


```
-----
+----+--------------------+
| key|               value|
+----+--------------------+
|NULL|{"user_id": 7372,...|
|NULL|{"user_id": 7373,...|
|NULL|{"user_id": 7374,...|
|NULL|{"user_id": 7375,...|
|NULL|{"user_id": 7376,...|
|NULL|{"user_id": 7377,...|
|NULL|{"user_id": 7378,...|
|NULL|{"user_id": 7379,...|
|NULL|{"user_id": 7380,...|
|NULL|{"user_id": 7381,...|
|NULL|{"user_id": 7382,...|
|NULL|{"user_id": 7383,...|
|NULL|{"user_id": 7384,...|
|NULL|{"user_id": 7385,...|
|NULL|{"user_id": 7386,...|
```

4. **Visualization**:
    - Streamlit dashboards for monitoring trends.


Action Counts
6
5
4
3
2
1
0
booked
canceled
viewed


Average Price by Room Type
500
400
300
200
100
0
Entire home/apt
Private room
Shared room



Technology Stack

Component	Tool Used	Rationale
Data Ingestion	Kafka	Real-time streaming platform with high throughput.
Storage	Cassandra	Combines batch storage with low-latency access.
Processing	Spark Streaming	Scalable and efficient for batch and streaming data.

Visualization	Streamlit	Lightweight and intuitive UI for live dashboards.
Programming	Python	Simple integration with all tools and libraries.

Data Workflow

Kafka → Spark → Cassandra → Streamlit.

1. A user books a listing.
2. The action is sent to a Kafka topic.
3. Spark processes the event and stores the structured data in Cassandra.
4. A Streamlit dashboard updates to reflect the new booking in real-time.
