

Pub/Sub to BigQuery – Implementation Summary

1. Pub/Sub Topic and Subscription

- A Pub/Sub topic named 'sainathbigquery_events' was created.
- A subscription named 'sainathbigquery-events-sub' was created for the topic.
- The subscription is active and able to receive messages.
- Messages published to the topic are successfully delivered to the subscription.

2. Publishing Messages to Pub/Sub

- User event messages are published in JSON format.
- Each message contains event type and user information.
- Example events include user login actions.
- Messages are successfully accepted by Pub/Sub.

3. Cloud Function Deployment

- A Cloud Function named 'sainathfunctionapp' was deployed in the asia-south1 region.
- The function is configured with a Pub/Sub trigger.
- It runs automatically whenever a message is published to the topic.

4. Message Processing in Cloud Function

- The Cloud Function decodes the Pub/Sub message.
- JSON data is extracted from the message payload.
- A timestamp is generated during processing.
- The processed data is prepared for storage in BigQuery.

[←](#) [sainath-event...](#) [Edit](#) [Create snapshot](#) [Replay messages](#) [Purge messages](#) [Show info pa](#)

Subscription name	projects/mazenet-001/subscriptions/sainath-events-sub
Subscription state	active
Topic name	projects/mazenet-001/topics/sainath-events
Tags	—

[Metrics](#) [Details](#) [Messages](#)

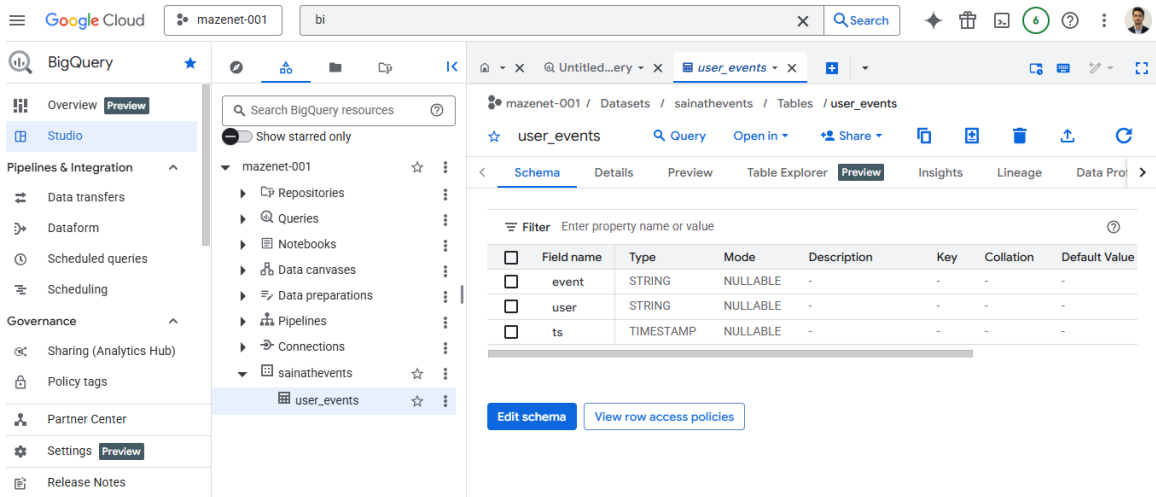
☒ Enable ack messages

Filter Filter messages

Publish time	Attribute keys	Message body	Ack ↑
Dec 23, 2025, 9:57:15 AM	—	{\event:\TestEvent\,name:\Sainath\,age:22,role:\admin\}	Deadline exceeded
Dec 23, 2025, 10:09:50 AM	—	{event: User_Created, user: sai, age: 23, role: data}	Deadline exceeded
Dec 23, 2025, 10:15:41 AM	—	{event: User_Created, user: Aarya, age: 30, role: admin}	Deadline exceeded

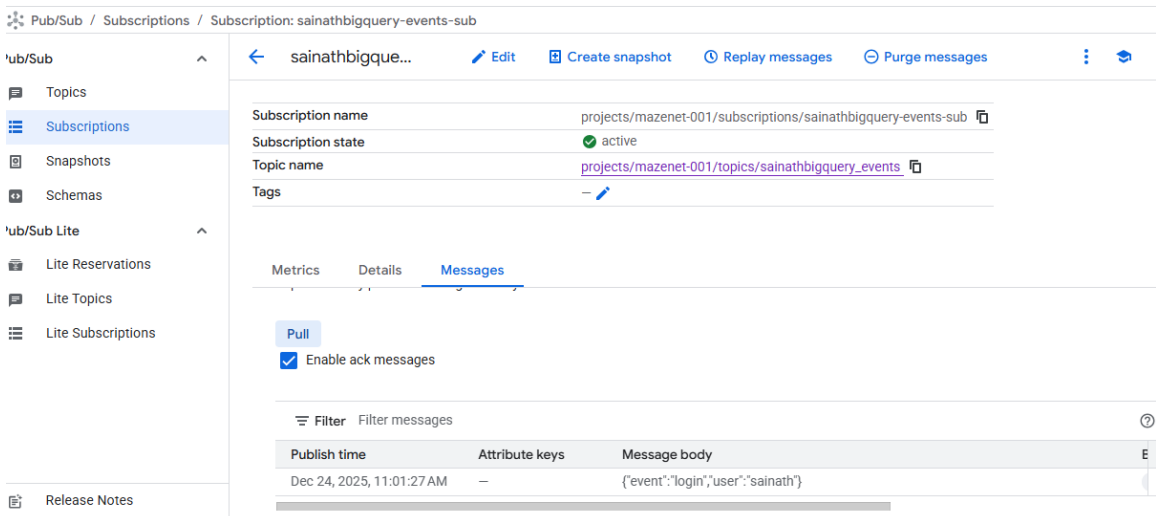
5. BigQuery Dataset and Table

- A BigQuery dataset named 'sainathevents' was created.
- A table named 'user_events' was created inside the dataset.
- Table schema includes event, user, and timestamp fields.



6. Data Insertion into BigQuery

- The Cloud Function inserts processed records into the BigQuery table.
- Multiple records are stored successfully.
- Each record contains event name, user name, and timestamp.



7. End-to-End Data Flow

- Events are published to Pub/Sub.
- Pub/Sub triggers the Cloud Function.
- The Cloud Function processes the data.
- BigQuery stores the final records for analysis.

Observability

Revisions

Source

Triggers

Networking

Security

YAML

Metrics

Logs

Severity

Default

Filter

Search all fields and values

?

Logs

Severity

Timestamp

Summary

>

i

2025-12-24 10:58:48.199 IST

Cloud Run ReplaceService sainathfunctionapp {@type: type.googleapis.com/google.cloud...

>

i

2025-12-24 11:01:29.469 IST

POST 200 130 B 690 ms APIs-Google; (+https://developers.goo... https://sainathfunctio...

>

*

2025-12-24 11:01:30.169 IST

Bigquery Insert Succeeded {'event': 'login', 'user': 'sainath', 'ts': '2025-12-24T05:31:2...

SLOs

No newer entries found matching current filter.

Errors

No newer entries found matching current filter.

Conclusion

This implementation demonstrates a real-time event processing system using Google Cloud Pub/Sub, Cloud Functions, and BigQuery. The solution is scalable, reliable, and suitable for real-time data analytics.