

Pub/Sub on GCP

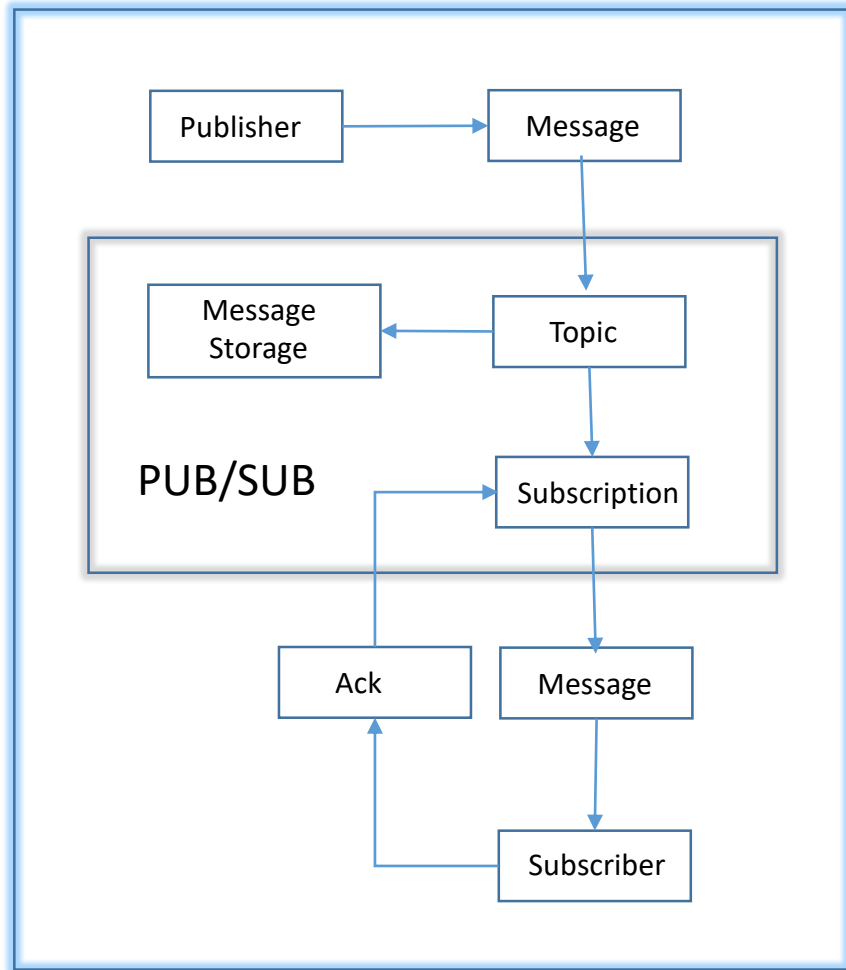
What is Pub/Sub?

- It is an asynchronous messaging service which can be used as a middleware for event based ingestion and streaming data loads.
- It features message durability and seamless performance at scale.
- It offers real time message delivery with high availability capabilities.

Concepts in Pub/Sub

- Topic: Entity where messages will be sent by publishers.
- Subscription: It represents the stream of messages from a specific topic, which will be delivered to the subscribing application.
- Message: The combination of data and attributes associated with it that a publisher sends to a topic which is delivered to subscribers.
- Message attribute: A key-value pair that a publisher can add to a message.

Pub/Sub Message Flow



1. A publisher creates a *topic* in the Pub/Sub and sends *messages* to the topic. A message contains data and attributes.
2. The pub/sub ensures message retention on behalf of subscriptions. A published message is retained for a subscription until it is acknowledged by any subscriber consuming messages from that subscription.
3. Pub/Sub forwards messages of a topic to all of its subscriptions, one by one.
4. A subscriber receives messages either by Pub/Sub *pushing* them to the subscriber's chosen endpoint, or by the subscriber *pulling* them from the service.
5. The subscriber sends an acknowledgement to the Pub/Sub service for each received message.
6. The acknowledged messages are removed from the subscription's message queue.

Pub/Sub vs Kafka

- **Fully Managed System** Both of these can have fully managed version in the cloud. Google provides Pub/Sub while Kafka is also offered as fully managed service by different cloud platforms.
- **Cloud vs On-prem** Pub/Sub is only available as a service in Google Cloud Platform , while Kafka is available as service and can also be configured on VM's manually.
- **Message duplication** - With Kafka Zookeeper is used to manage the offsets of the messages and this gives flexibility to track the messages read so far by the Consumers. Pub/Sub works using acknowledging the message, if your code doesn't acknowledge the message before the deadline, the message is sent again, that way you can avoid duplicated messages.
- **Retention policy** Both have options to configure the maximum retention time, by default is 7 days.
- **Consumers Group vs Subscriptions** Pub/Sub use subscriptions, you create a subscription and then you start reading messages from that subscription. Once a message is read and acknowledge, the message for that subscription is gone. Kafka use the concept of "consumer group" and "partition", every consumer process belongs to a group and when a message is read from a specific partition, then any other consumer process which belongs to the same "consumer group" will not be able to read that message (that is because the offset eventually will increase).