Pubsub

Pub/Sub in GCP

Google Cloud Pub/Sub is a fully managed, real time messaging service used to send**,** receive, and process events asynchronously.

It follows a **publish–subscribe model**:

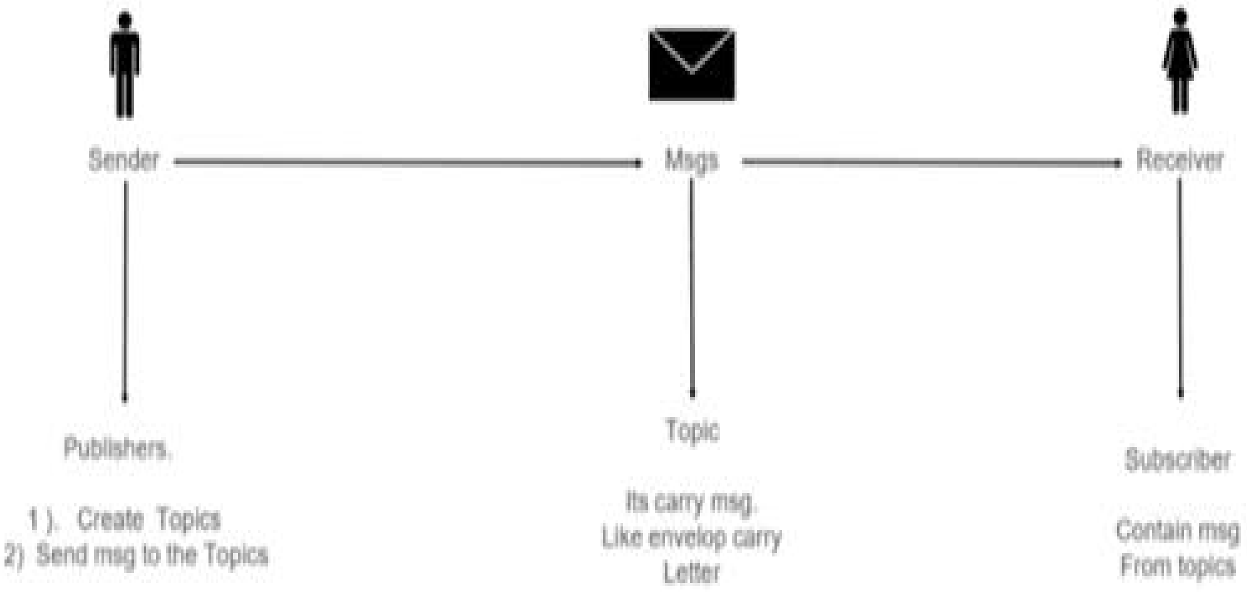
Publisher → sends messages

Topic → holds messages

Subscriber → receives messages

Publishers and subscribers do not know about each other.

How Pub/Sub works (simple flow)



1. Publisher sends message to Topic

   A publisher application sends an event/message to a Pub/Sub topic.The publisher does not know or care who will consume the message.

2. Message is stored durably

   Pub/Sub safely stores the message across Google’s infrastructure.This ensures the message is not lost even if subscribers are temporarily down

3. Subscribers pull or receive pushed messages

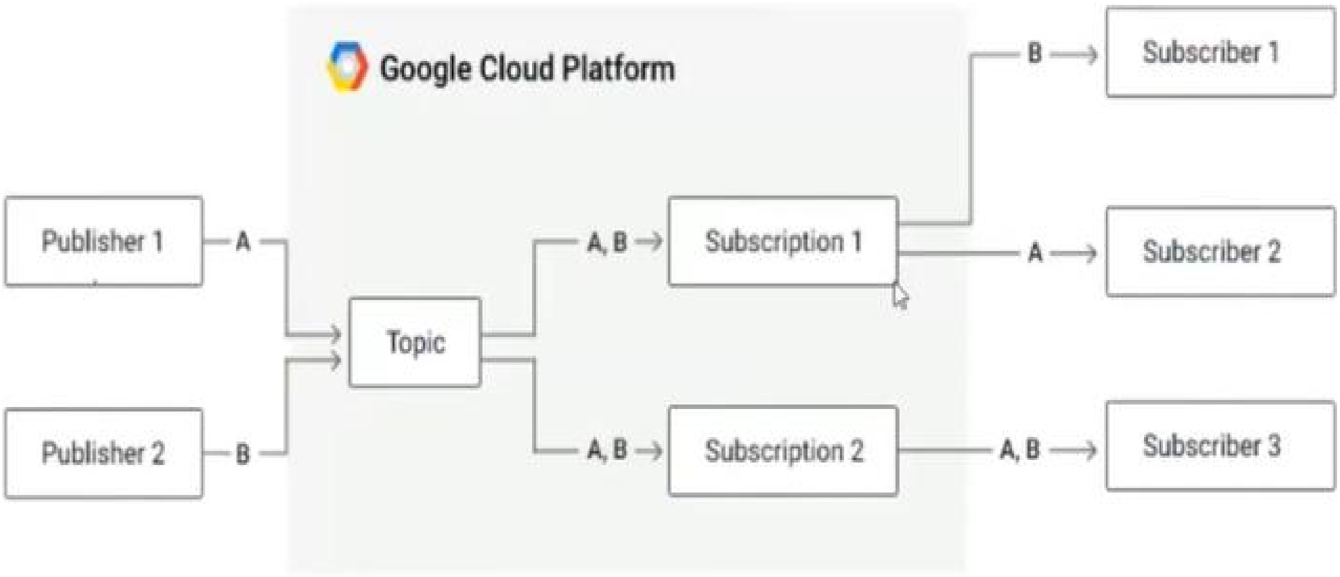
   Subscribers receive messages either by pulling them or via push delivery.Multiple subscribers can receive the same message independently.

4. Subscriber acknowledges message

   After processing, the subscriber sends an acknowledgment to Pub/Sub.This tells Pub/Sub the message was successfully handled

5. If not acknowledged → message is redelivered

If acknowledgment is not received within the time limit, Pub/Sub retries.
This guarantees at-least-once delivery of every message.



Key components

Topic: Logical channel for messages

Publisher: Application that sends messages

Subscription: Connection between topic and subscriber

Subscriber: Application that consumes messages

Types of subscriptions

Pull: Subscriber pulls messages manually

Push: Pub/Sub pushes messages to an endpoint (HTTP, Cloud Run)

Benefits of using Pub/Sub

1. Fully managed

   No servers

   No maintenance

2. Massive scalability

   Handles millions of messages per second

3. Loose coupling

   Services can evolve independently

4. Reliable delivery

   At-least-once delivery

   Automatic retries

5. Global & low latency

Works across regions

6. Cost-effective

Pay only for usage

7. Easy integration

Native integration with:

Dataflow, Cloud Functions, Cloud Run, BigQuery

Where do we use Pub/Sub?

1. Event-driven architectures

Example:

User uploads a file

Event is published

Multiple services react:

- Image processing
- Metadata extraction
- Notification service

2. Data ingestion pipelines

Example:

Streaming data from:

- IoT devices
- Logs
- Clickstream data
- Pub/Sub → Dataflow → BigQuery

3. Microservices communication

Example:

Order service publishes order_created

Inventory, billing, and email services consume it independently

Why do we use the pubsub:

> Decouple systems: Services communicate through events instead of direct calls, so they don’t depend on each other.

> Handle real-time data: Events are delivered and processed immediately as soon as they occur.

> Process events asynchronously: The producer sends a message and continues its work without waiting for consumers.

> Scale automatically: Pub/Sub handles traffic spikes by buffering messages and scaling consumers independently.

Pratical workflow

Creation of topic :

```
> gcloud pubsub topics create mytopic
```

Creation of subscription:

```
>gcloud pubsub subscriptions create --topic mytopic
mysub
```

Pass a message as event :

```
> gcloud pubsub topics publish mytopic --message
“ Hello Team “
```

- defines the message id

Pull the message

```
> gcloud pubsub subscriptions pull --auto-ack mysub
```

- message is displayed in the subscription