|  |  |
| --- | --- |
| Working with Pub-Sub | Understanding Pub-Sub |
| Creating/Subscribing to topic |

Google Cloud Pub/Sub (Publisher-Subscriber) is a messaging service designed to allow communication between different applications. It's a powerful tool that facilitates the decoupling of your application components, enabling them to communicate asynchronously. Let's break down Pub/Sub in detail, from its basic concepts to creating and subscribing to topics:

### 1. \*\*Understanding Pub/Sub:\*\*

\*\*Pub/Sub Model:\*\*

- Pub/Sub follows a publisher-subscriber model.

- Publishers send messages to a central system (the topic), and subscribers receive messages from that topic.

- This asynchronous communication model enables the decoupling of the sender (publisher) from the receiver (subscriber).

\*\*Basic Concepts:\*\*

- \*\*Topic:\*\*

  - A named resource to which messages are sent by publishers.

  - A topic is like a mailbox—messages are sent to the topic's name, and subscribers pull messages from the same name.

  - Think of it as a channel through which you communicate.

- \*\*Subscription:\*\*

  - Represents the stream of messages from a single, specific Pub/Sub topic.

  - Subscriptions are named resources that represent the stream of messages from a topic.

  - You can have multiple subscriptions for a single topic, each of which acts as an independent receiver.

- \*\*Message:\*\*

  - The data that gets sent from a publisher to a subscriber via a topic.

  - A message consists of two parts: the data payload and attributes.

  - Payload: The data you want to send.

  - Attributes: Key-value pairs that can contain metadata.

\*\*Key Benefits:\*\*

- \*\*Scalability:\*\*

  - Pub/Sub can handle high throughput and scales seamlessly.

- \*\*Reliability:\*\*

  - Messages persist in Pub/Sub, ensuring reliable delivery.

- \*\*Flexibility:\*\*

  - Supports both synchronous and asynchronous message processing.

- \*\*Decoupling:\*\*

  - Allows decoupling of services, making systems more flexible and easier to develop and scale.

### 2. \*\*Creating/Subscribing to a Topic:\*\*

\*\*Creating a Topic:\*\*

1. \*\*Using the Cloud Console:\*\*

   - Go to the Pub/Sub section of the Cloud Console.

   - Click "Create Topic."

   - Give your topic a name and click "Create."

2. \*\*Using the Command Line:\*\*

**```bash**

**gcloud pubsub topics create [TOPIC\_NAME]**

**```**

\*\*Subscribing to a Topic:\*\*

1. \*\*Using the Cloud Console:\*\*

   - Go to the Pub/Sub section of the Cloud Console.

   - Click on the topic you want to subscribe to.

   - Click "Create Subscription."

   - Give your subscription a name.

   - Configure the settings for the subscription.

   - Click "Create."

## Message retention duration

 Duration is from 10 minutes to 7 days

## Acknowledgement deadline

Deadline time is from 10 seconds to 600 seconds

2. \*\*Using the Command Line:\*\*

   ```bash

**gcloud pubsub subscriptions create [SUBSCRIPTION\_NAME] --topic [TOPIC\_NAME]**

   ```

\*\*Publishing a Message to a Topic:\*\*

1. \*\*Using the Cloud Console:\*\*

   - Go to the Pub/Sub section of the Cloud Console.

   - Click on the topic you want to publish to.

   - Click "Publish Message."

   - Enter the message payload and any attributes.

   - Click "Publish."

2. \*\*Using the Command Line:\*\*

   ```bash

   gcloud pubsub topics publish [TOPIC\_NAME] --message "[MESSAGE]"

   ```

\*\*Subscribing to Receive Messages:\*\*

1. \*\*Using the Cloud Console:\*\*

   - Go to the Pub/Sub section of the Cloud Console.

   - Click on the subscription you want to receive messages from.

   - Click "Pull."

   - Messages will be displayed, and you can acknowledge them.

2. \*\*Using the Command Line:\*\*

   ```bash

**gcloud pubsub subscriptions pull --auto-ack [SUBSCRIPTION\_NAME]**

   ```

While creating topic options

**Add a default subscription**

Subs Automatic creates( default attach subs to topic) when the topic is created

**Use a schema**

A schema is a format that messages from a topic must follow. When creating a topic you can choose to create a new schema to assign to it, or assign it an existing

**Enable ingestion**

When enabled, streams data into the topic from an external source.

**Enable message retention**

When enabled, retains messages on the Topic for up to 31 days.

**Backup message data to Cloud Storage**