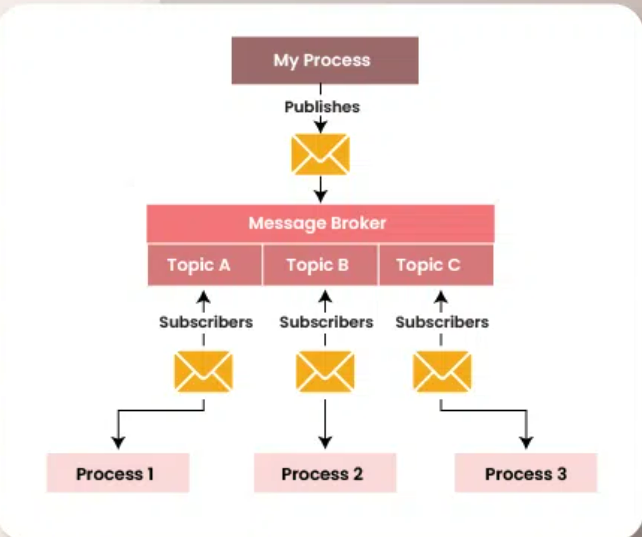
Publisher and Subscriber

## What is Pub / Sub?

It is a messaging pattern. It is used in software architecture.

**Publisher:**

The publisher will generate and send messages to the centralized store or message broker without bothering about the receiver’s availability.

**Subscriber:**The subscriber component is used to receives message from the centralized location or message broker. Also, it can receive messages from a specific source.  
  


**Message Broker:** Here the message broker will act like an **interceptor** between the Pub/Sub.  
  
 **Topics:** It is a named **channels** or **categories** to which messages are published.

**Use Case:**

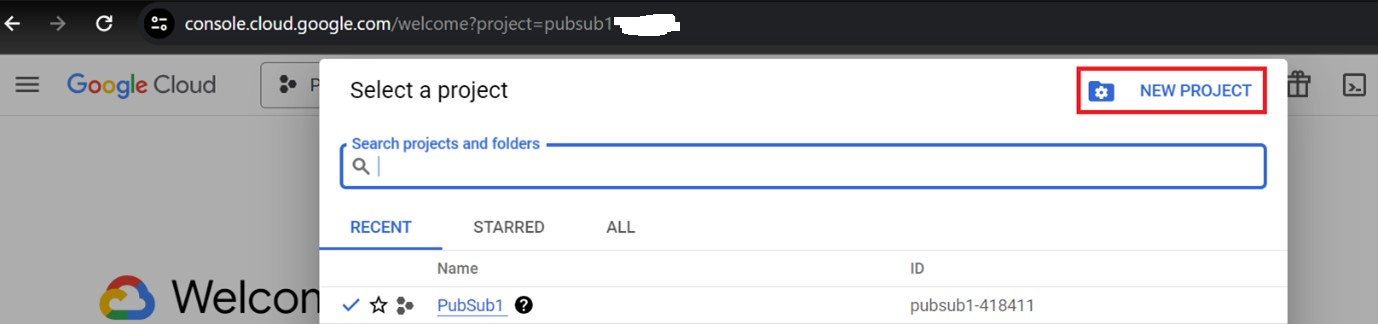
The pub/sub are used in the scenarios where **asynchronous and scalable communication between components is required**.

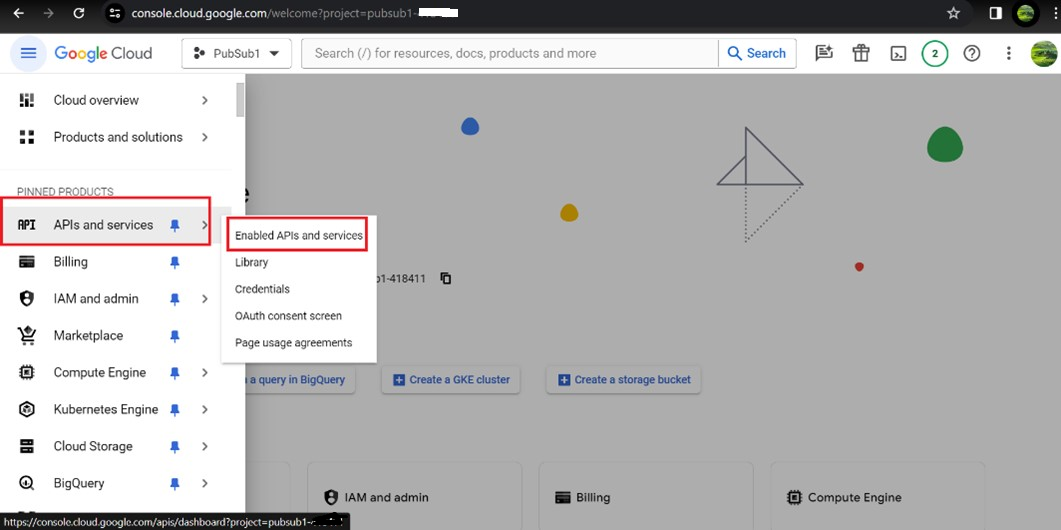
1. Real-time Data Streaming – sensor devices and Biometric devices (the device will publish the data stream for multiple users simultaneously)
2. Event-driven Architectures – Instead of making API Call to the server. It allows components to subscribe to specific events and receive notifications when those events occur, enabling more responsive and reactive systems.
3. Message Queues
4. Notifications and Alerts
5. Scalable Web Applications
6. Microservices Communication

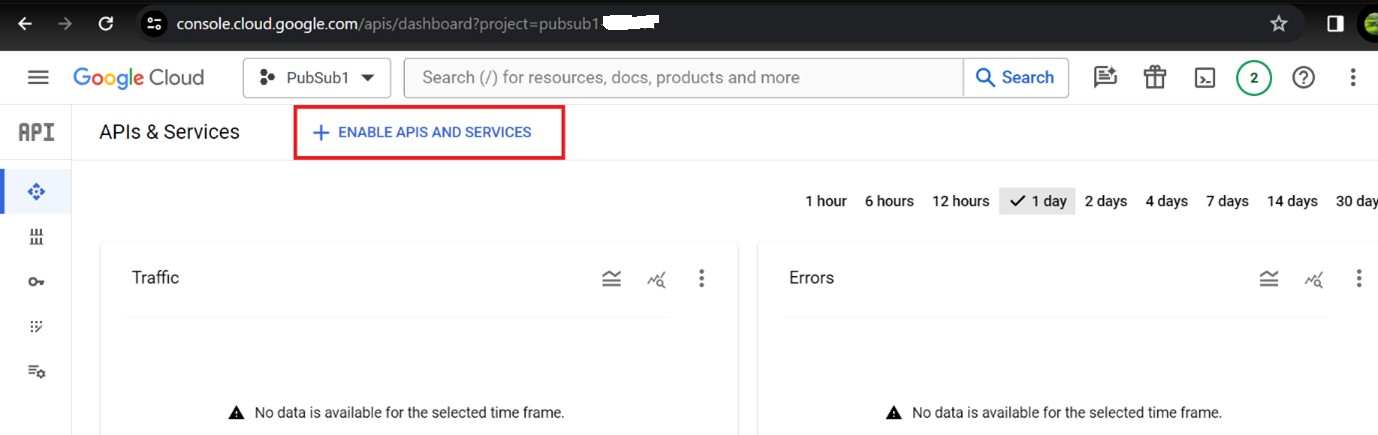
**Steps to publish a message in Google Cloud Console:  
  
Step-1:**Navigate to Google Cloud Console: Open your web browser and go to the Google Cloud Console at <https://console.cloud.google.com/>

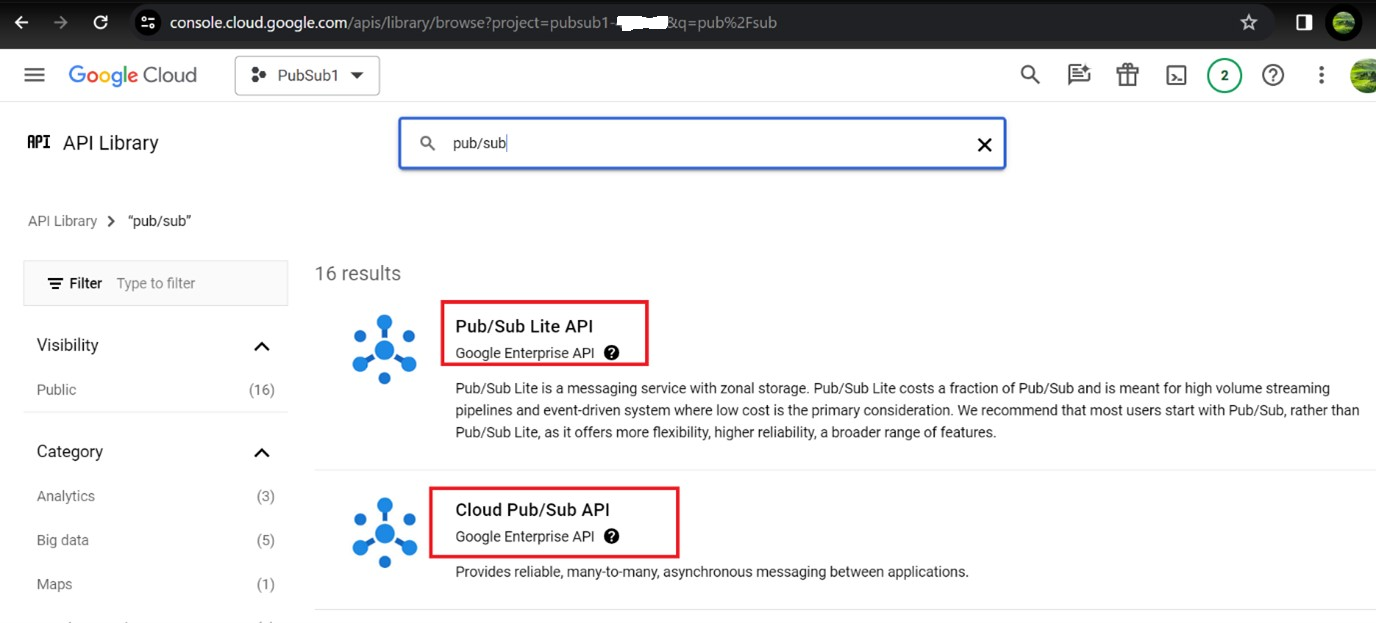
**Step-2**:

Select or **Create a Project**: If you haven't already created a project, you'll need to create one. Otherwise, select the project where you want to work with Pub/Sub.



**Step-3:**Once the project created successfully. Enable the Pub/Sub API for your project.  
  


****

Note: you can use either lite version or regular version  
  


**Step-4:**

**Create a Topic (if necessary):** If you haven't already created a topic to publish messages to, you'll need to create one. Click on "Create Topic" and give your topic a name. Click "Create" to confirm.

**Step-5:**

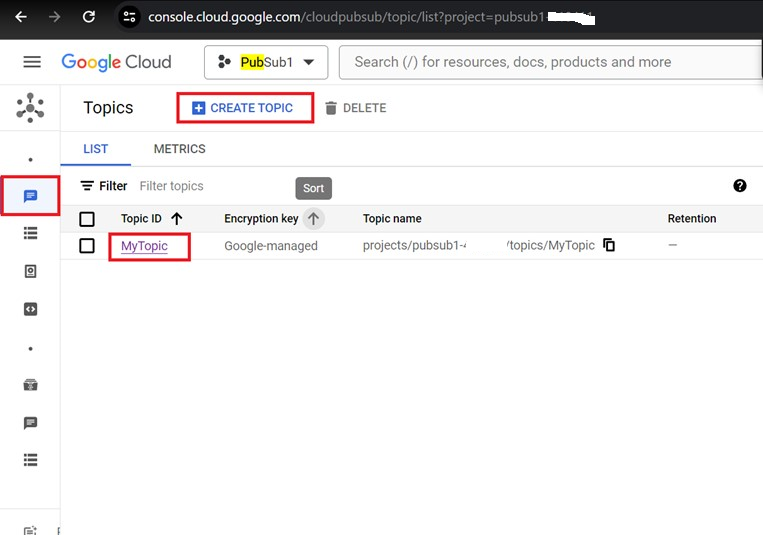
**Steps to create a topic:**

1. In the Google Cloud console, go to the Pub/Sub page.

[Go to Pub/Sub](https://console.cloud.google.com/cloudpubsub/topicList?_ga=2.176842021.1429380309.1711428959-916947591.1711428959)

1. In the Topics page, click Create topic.
2. In the window that opens, enter MyTopic in the Topic ID field.
3. Leave the default values for the remaining options, and then click Create.

You see the success message: A new topic and a new subscription have been successfully created.

You have just created a topic called MyTopic and an associated default subscription MyTopic-sub.  
  


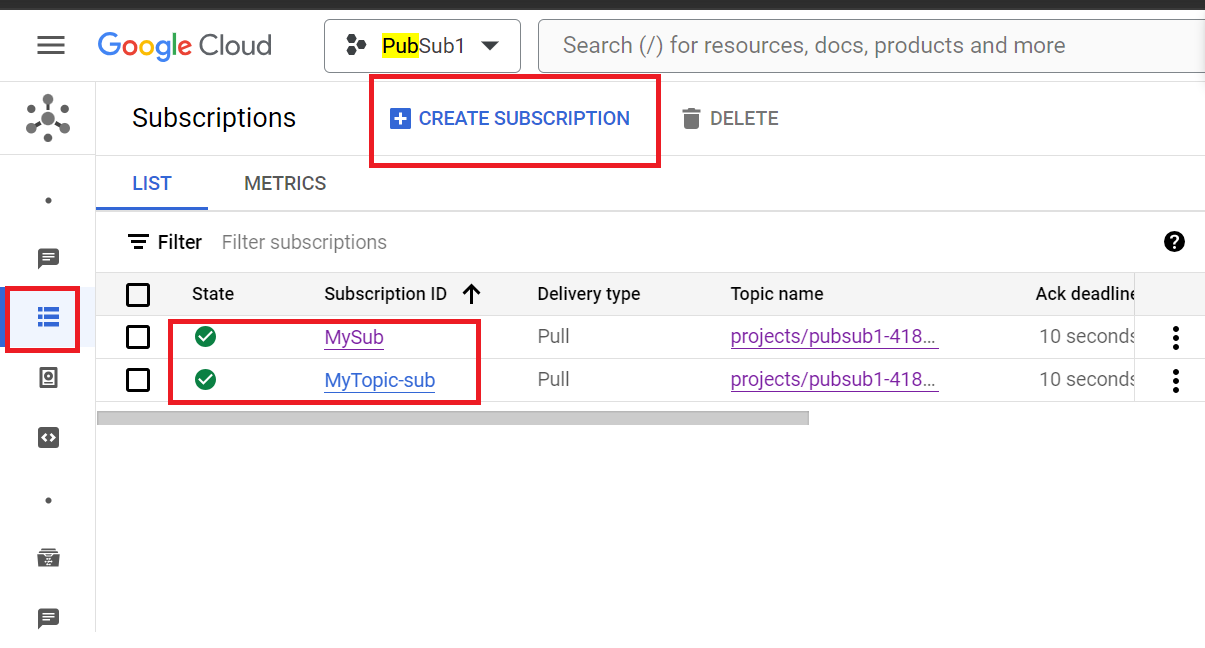
**Add a second subscription:**

To add a second subscription to the topic you just created, complete these steps:

1. Click Subscriptions.
2. In the Subscriptions page, click Create subscription.
3. Enter MySub in the Subscription ID field.
4. For Select a Cloud Pub/Sub topic, select the MyTopic topic from the drop-down menu.
5. Leave the default values for the remaining options.
6. Click Create.

You see the success message: Subscription successfully added.

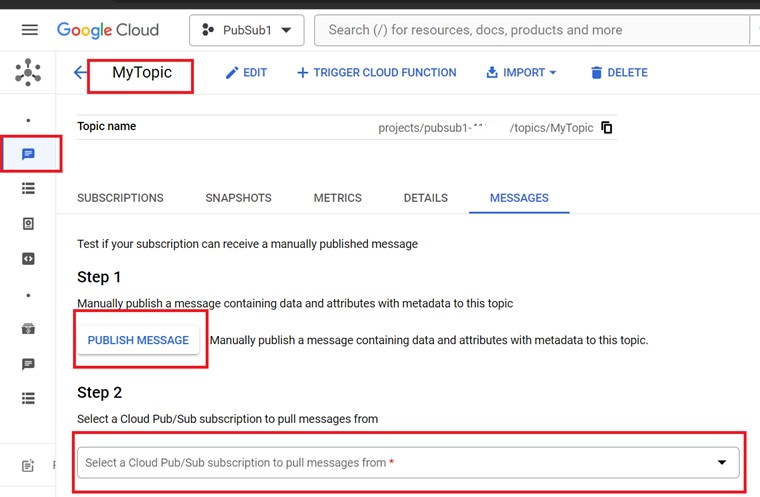
1. Click the Topics page and click MyTopic.

The MySub subscription is now attached to the topic MyTopic. Pub/Sub delivers all messages sent to MyTopic to the MySub and MyTopic-sub subscriptions.  
  


**Publish a message to the topic:**

1. In the Google Cloud console, go to the Pub/Sub topics page.

[Go to Topics](https://console.cloud.google.com/cloudpubsub/topicList)

1. Click the MyTopic topic.
2. In the MyTopic page, click the Messages tab.
3. Click Publish message.
4. In the Message body window, enter Hello World.
5. Click Publish. A message displays at the bottom of the page that says "Message published" if the publish was successful.  
     
   

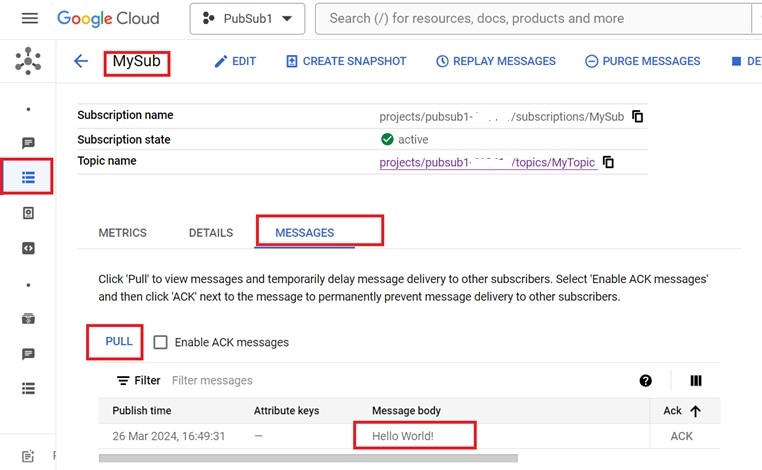
**Pull the messages from the subscription:**

1. In the Google Cloud console, go to the Pub/Sub subscriptions page.

[Go to Subscriptions](https://console.cloud.google.com/cloudpubsub/subscription/list?_ga=2.181019559.1429380309.1711428959-916947591.1711428959)

1. In the Messages tab, click Pull.

You should see the message that you just published. The message has the data, Hello World, and the time when the message was published.

When using the Google Cloud console, an individual pull for a low message volume can often return zero messages. If you do not see messages, click Pull multiple times to issue multiple pull requests. This is not an issue with the [Pub/Sub Client Libraries](https://cloud.google.com/pubsub/docs/publish-receive-messages-client-library).  
  


**Clean up:**

To avoid incurring charges to your Google Cloud account for the resources used on this page, follow these steps.

If you no longer need the topic and subscription you created, you can delete them.

To delete the topic and subscription, complete these steps:

1. Go to the Topics page.
2. Check the checkbox next to MyTopic and click Delete.
3. Complete the steps in the confirmation window to permanently delete the topic.

Or you can [delete the project](https://console.cloud.google.com/cloud-resource-manager) that you created.

**How to fetch the created pub / sub to the Frontend or UI?:** 1. **Set up Google Cloud Pub/Sub:**

* + Create a Google Cloud project if you haven't already.
  + Enable the Pub/Sub API for your project.
  + Create a Pub/Sub topic where messages will be published.
  + Set up Pub/Sub subscriptions if you want to receive messages.

1. **Install the necessary dependencies:**

You'll need to use Google Cloud client libraries for JavaScript to interact with Pub/Sub from your React application. You can install these dependencies using npm or yarn.



1. **Authentication**:

You need to authenticate your React application to access Google Cloud services. The recommended way to do this is by using service account credentials. Create a service account in your Google Cloud project and download the JSON key file. Make sure to securely store this key file and never expose it publicly.

1. **Integrate with React**:

* Import the Pub/Sub client library in your React component where you want to use Pub/Sub.
* Use the client library to publish messages to the Pub/Sub topic.

1. **Example Code**: Here's a basic example of how you might integrate Pub/Sub with a React component to publish messages:

**A computer screen shot of a program

Description automatically generated**

**Handling Authentication**: You'll need to handle authentication in your React application. This might involve securely storing your service account key file and managing access tokens for authentication.  
  
Official guide:  
<https://cloud.google.com/nodejs/docs/reference/pubsub/latest>

<https://github.com/googleapis/nodejs-pubsub>