Experiment-07

Cloud Pub/Sub: Experiment how Google Cloud Pub/Sub facilitate real-time messaging and communication between distributed applications

Google Cloud Pub/Sub is a fully managed **messaging service** that enables **asynchronous, real-time communication** between distributed applications. It follows a **publish-subscribe** model where **publishers send messages to topics** and **subscribers receive them** via push or pull delivery.

Why Use Cloud Pub/Sub?

Real-time messaging – Delivers messages instantly across services.

Decouples components – Microservices can communicate asynchronously.

High availability & scalability – Handles millions of messages per second.

Guaranteed delivery – Retries messages until they are acknowledged.

Security – Integrated with **IAM** for access control.

Cloud Pub/Sub Architecture

Publisher – Sends messages to a **Topic**.
Topic – A named channel where messages are published.
Subscription – Defines how messages are delivered to subscribers.
Subscriber – Reads messages from a subscription.
Delivery Mechanism – **Pull** (manual retrieval) or **Push** (automatic HTTP delivery).

Step-by-Step: Experimenting with Cloud Pub/Sub

Step 1: Enable Cloud Pub/Sub API

1. Open **Google Cloud Console → Navigation Menu (☰) → APIs & Services → Library**.
2. Search for **Cloud Pub/Sub API** and click **Enable**.

Step 2: Create a Pub/Sub Topic

1. Go to **Navigation Menu (☰) → Pub/Sub → Topics**.
2. Click **Create Topic**.
3. Enter a **Topic ID** (e.g., my-topic).
4. Click **Create**.

Your topic is now ready!

Step 3: Create a Subscription

1. Click on your **Topic → Create Subscription**.
2. Enter a **Subscription ID** (e.g., my-subscription).
3. Choose a **Delivery Type**:
    - **Pull** – Messages are manually fetched by the subscriber.
    - **Push** – Messages are automatically sent to an HTTP endpoint.

4. Click **Create**.

Your subscription is now linked to the topic!

Step 4: Publish a Message (Using gcloud CLI)

Run the following command in **Cloud Shell**:

gcloud pubsub topics publish my-topic --message "Hello, Pub/Sub!"

Message published successfully!

Step 5: Pull Messages from Subscription (Using gcloud CLI)

Run the following command:

gcloud pubsub subscriptions pull my-subscription --auto-ack

This will **retrieve and acknowledge messages** from my-subscription.

You will see the message: Hello, Pub/Sub!

Step 6: Publish & Subscribe Using Python (Optional)

Install Google Cloud Pub/Sub SDK

pip install google-cloud-pubsub

Publisher Code (Python)

from google.cloud import pubsub_v1

```
project_id = "your-project-id"

topic_id = "my-topic"


publisher = pubsub_v1.PublisherClient()

topic_path = publisher.topic_path(project_id, topic_id)


message = "Hello, Pub/Sub from Python!"

future = publisher.publish(topic_path, message.encode("utf-8"))

print(f"Published message ID: {future.result()}")
```

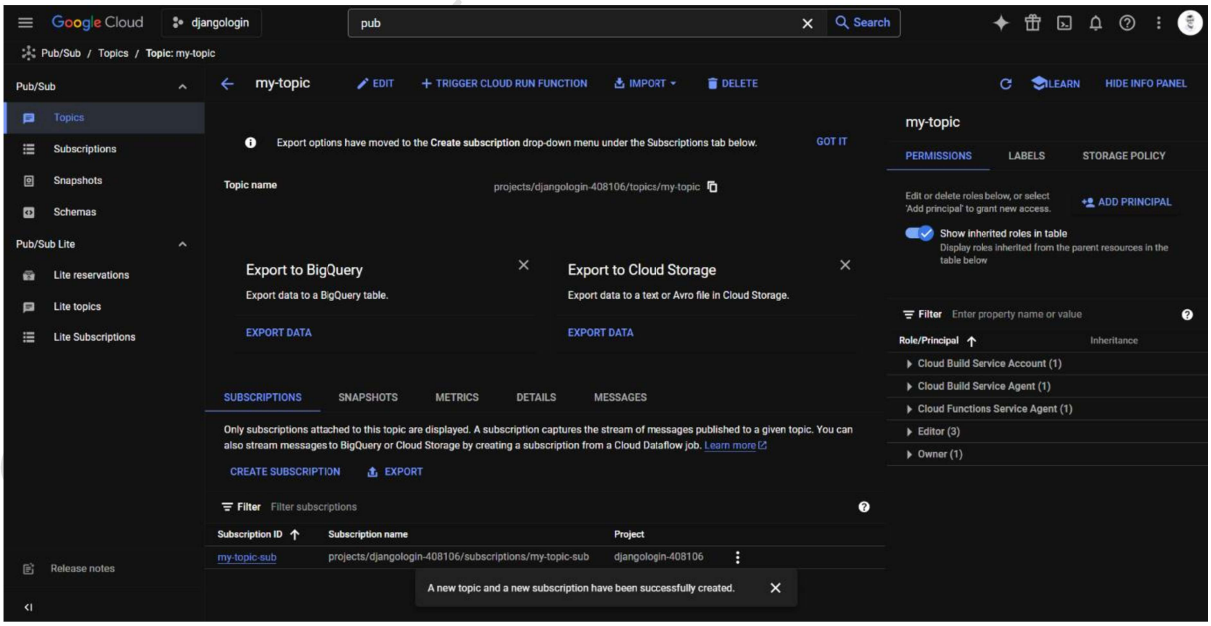
Subscriber Code (Python)

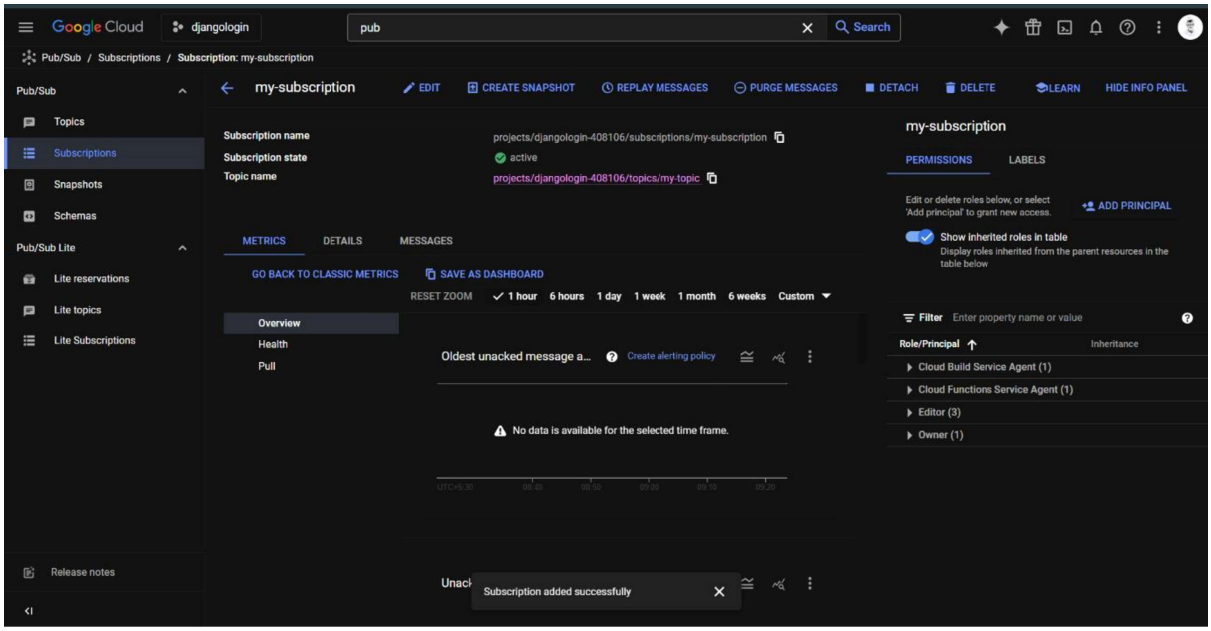
```
from google.cloud import pubsub_v1


project_id = "your-project-id"

subscription_id = "my-subscription"


subscriber = pubsub_v1.SubscriberClient()

subscription_path = subscriber.subscription_path(project_id, subscription_id)


def callback(message):

    print(f"Received: {message.data.decode('utf-8')}")
```

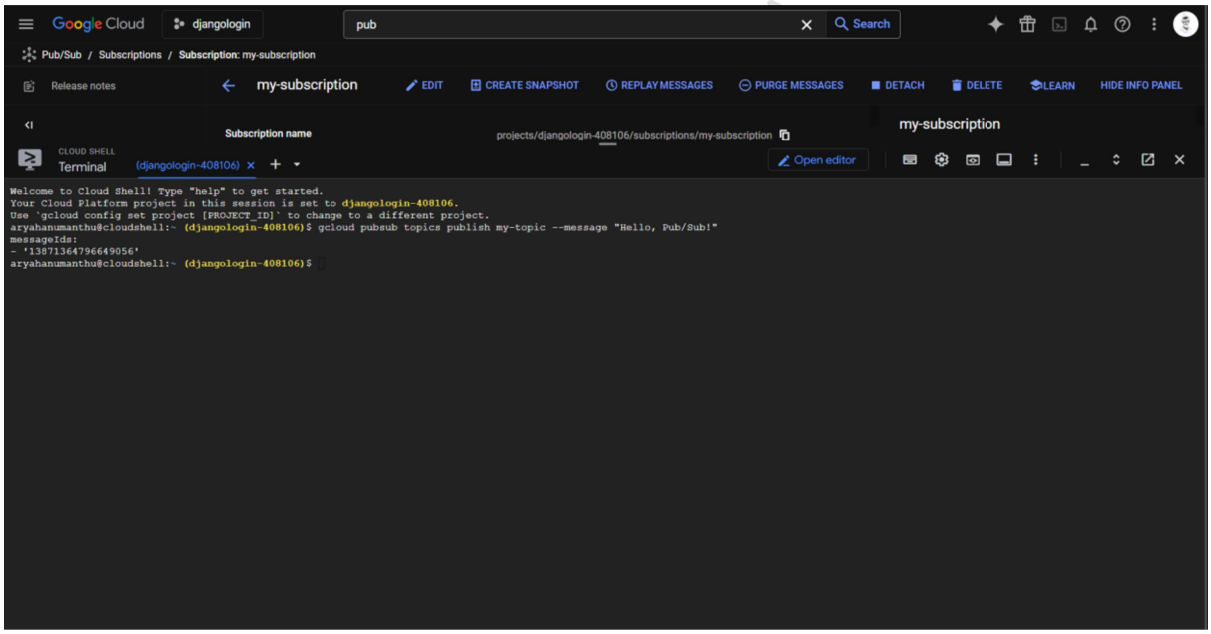
```
    message.ack()


subscriber.subscribe(subscription_path, callback=callback)

print("Listening for messages...")


import time

while True:

    time.sleep(10)
```

Now, whenever you publish a message, the subscriber will receive it in real-time!

Output




Google Cloud
djangologin
pub
Search
Pub/Sub / Subscriptions / Subscription: my-subscription
Pub/Sub
Topics
Subscriptions
Snapshots
Schemas
Pub/Sub Lite
Lite reservations
Lite topics
Lite Subscriptions
Release notes
my-subscription
EDIT
CREATE SNAPSHOT
REPLAY MESSAGES
PURGE MESSAGES
DETACH
DELETE
LEARN
HIDE INFO PANEL
Subscription name
projects/djangologin-408106/subscriptions/my-subscription
Subscription state
active
Topic name
projects/djangologin-408106/topics/my-topic
METRICS
DETAILS
MESSAGES
GO BACK TO CLASSIC METRICS
SAVE AS DASHBOARD
RESET ZOOM
1 hour
6 hours
1 day
1 week
1 month
6 weeks
Custom
Overview
Health
Pull
Oldest unacked message a...
Create alerting policy
No data is available for the selected time frame.
Subscription added successfully
my-subscription
PERMISSIONS
LABELS
Edit or delete roles below, or select 'Add principal' to grant new access.
ADD PRINCIPAL
Show inherited roles in table
Display roles inherited from the parent resources in the table below
Filter
Enter property name or value
Role/Principal
Inheritance
Cloud Build Service Agent (1)
Cloud Functions Service Agent (1)
Editor (3)
Owner (1)


Google Cloud
djangologin
pub
Search
Pub/Sub / Subscriptions / Subscription: my-subscription
Release notes
my-subscription
EDIT
CREATE SNAPSHOT
REPLAY MESSAGES
PURGE MESSAGES
DETACH
DELETE
LEARN
HIDE INFO PANEL
Subscription name
projects/djangologin-408106/subscriptions/my-subscription
CLOUD SHELL
Terminal
(djangologin-408106)
Open editor
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to djangologin-408106.
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
aryahanumanthu@cloudshell:~ (djangologin-408106)$ gcloud pubsub topics publish my-topic --message "Hello, Pub/Sub!"
messageIds:
- '13871364796649056'
aryahanumanthu@cloudshell:~ (djangologin-408106)$