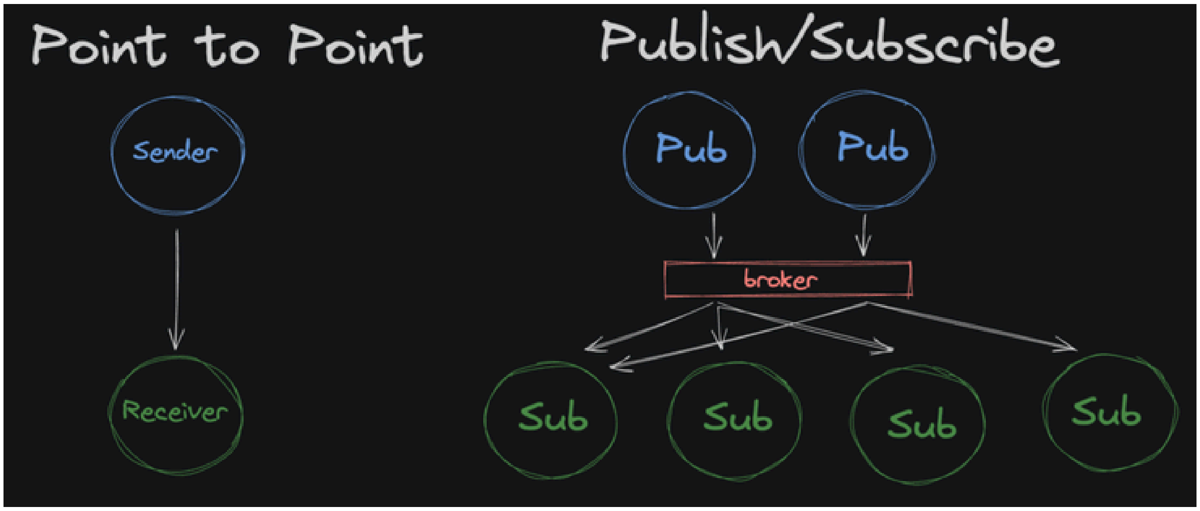
01 PubSub intro

Pub/Sub (publish/subscribe) is a pattern software systems can use to communicate. It enables architectures that can be more:

- Scalable
- Reliable
- Maintainable

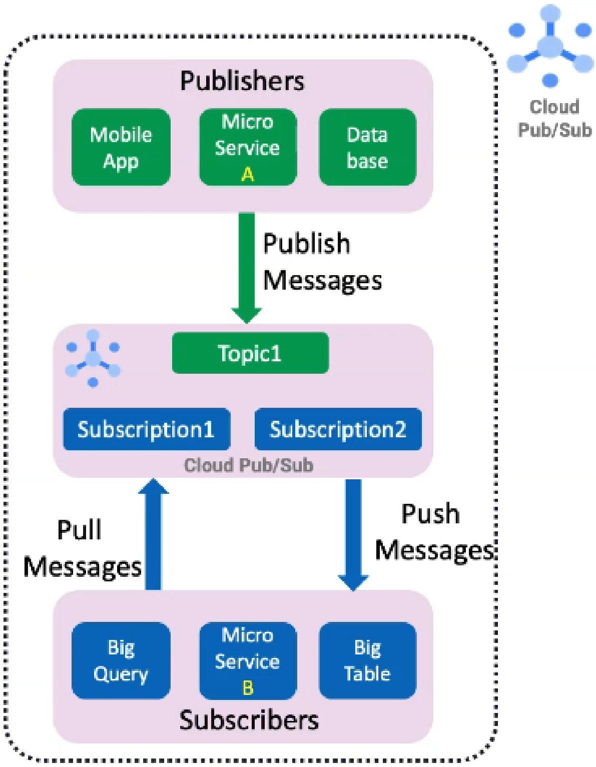
Pub/Sub systems are the opposite of point-to-point systems. In point-to-point systems, the sender is painfully aware of the receiver. For example, an HTTP request is point-to-point. The client sends a request *directly* to the server, and the server sends a response *directly* back to the client.
Email - point-to-point
YouTube: pub/sub system



Twitter (X) works like a Pub/Sub system. When you tweet, you don't individually list all of the followers you want to send your tweet to. Instead, you just tweet, and Twitter's servers deliver your tweet to all of your followers.

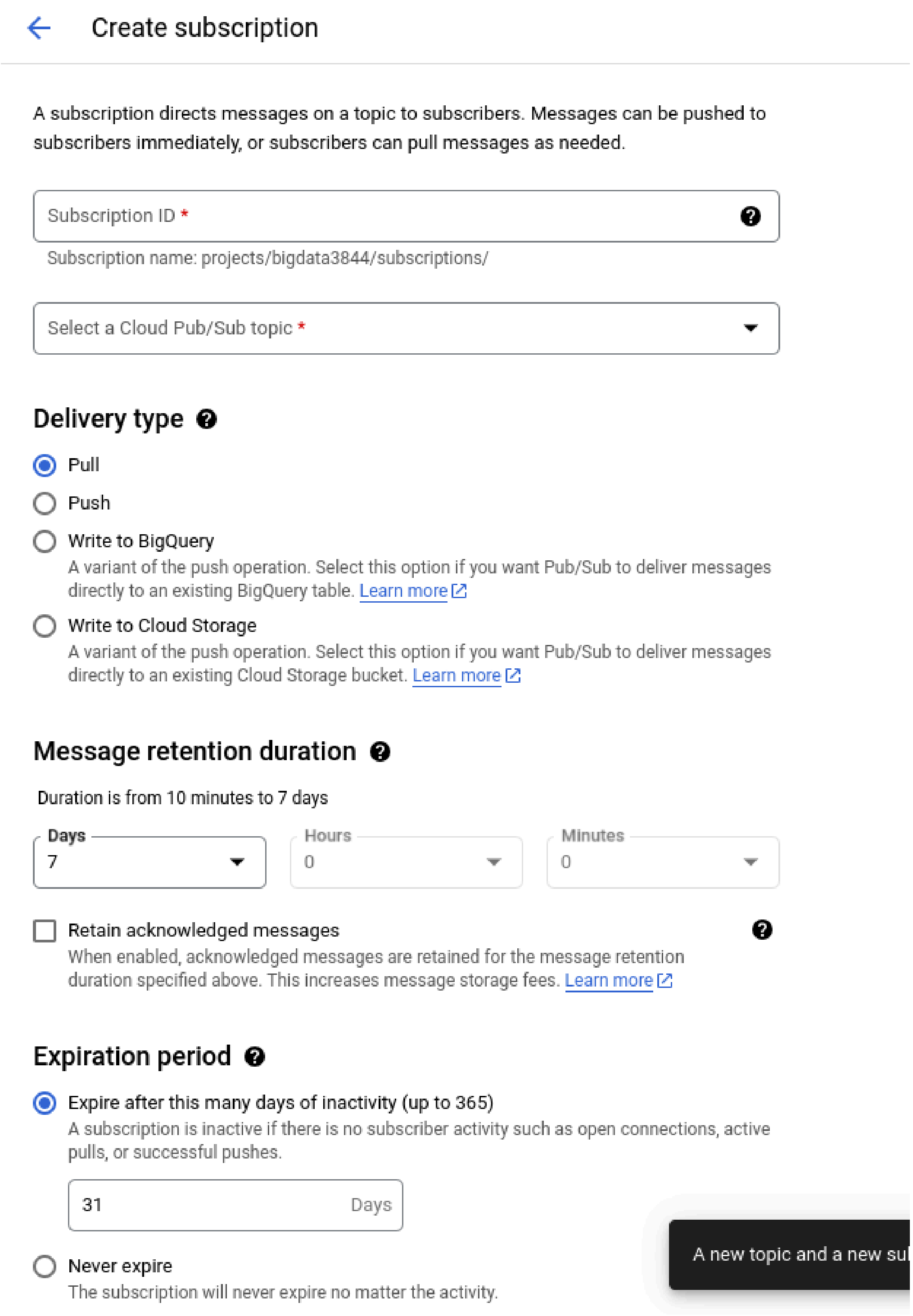
Google Cloud Pub/Sub

- Cloud Pub/Sub: Pub/Sub is a fully-managed asynchronous messaging service designed to be highly reliable and scalable.
- Google products, such as Ads, Search, and Gmail, send 500 million messages per second using this infrastructure, totaling over 1TB/s of data.
- Primarily used for real-time data streaming and event-driven systems.
- Stream Analytics (real-time data streaming)
  - Very powerful feature
  - Ingest analytic events of our applications and stream them to BigQuery, with Dataflow.



- **Publishers**
  - Services that produce messages
  - Publishers send events to Pub/Sub Topics, without worrying about when or how these events will be handled.

- **Subscribers**
  - Services that process those messages
  - Subscribers subscribe to a Pub/Sub subscription
  - Subscriptions have the following delivery types:
    - **Pull**: Subscriber needs to pull the messages
    - **Push**,
    - **Write to BigQuery**,
    - **Write to Cloud Storage**
  - As soon as a message arrives at a Pub/Sub topic, the subscription will push them to registered subscribers.



← Create subscription

A subscription directs messages on a topic to subscribers. Messages can be pushed to subscribers immediately, or subscribers can pull messages as needed.

Subscription ID *

Subscription name: projects/bigdata3844/subscriptions/

Select a Cloud Pub/Sub topic *

Delivery type

● Pull

○ Push

○ Write to BigQuery
A variant of the push operation. Select this option if you want Pub/Sub to deliver messages directly to an existing BigQuery table. Learn more

○ Write to Cloud Storage
A variant of the push operation. Select this option if you want Pub/Sub to deliver messages directly to an existing Cloud Storage bucket. Learn more

Message retention duration

Duration is from 10 minutes to 7 days

Days: 7
Hours: 0
Minutes: 0

□ Retain acknowledged messages
When enabled, acknowledged messages are retained for the message retention duration specified above. This increases message storage fees. Learn more

Expiration period

● Expire after this many days of inactivity (up to 365)
A subscription is inactive if there is no subscriber activity such as open connections, active pulls, or successful pushes.

31 Days

○ Never expire
The subscription will never expire no matter the activity.

A new topic and a new su

- **Global Service**
  - Pub/Sub is a global service
  - Topics and subscriptions are not region-specific
  - Messages flow within the Pub/Sub service between regions when needed
  - When using the global endpoint (`pubsub.googleapis.com`), publishers and subscribers connect to the nearest network region where Pub/Sub runs.
  - When using the locational endpoints (`us-central1-pubsub.googleapis.com`), publishers and subscribers connect to Pub/Sub in the specified region.
- **Autoscaling**

  - Designed to scale horizontally
  - No provisioning, not visible, everything happens in the background
  - Auto-everything

Life of a message

- A publisher sends a message to Pub/Sub.

- The message is written to Pub/Sub storage.

- Pub/Sub sends an acknowledgement to the publisher that it has received the message and guarantees its delivery to all attached subscriptions.

- At the same time as writing the message to storage, Pub/Sub delivers it to subscribers.

- Subscribers send an acknowledgement to Pub/Sub that they have processed the message.

- Once at least one subscriber for each subscription has acknowledged the message, Pub/Sub deletes the message from storage.

- **Compliance and Security**
  - HIPAA-compliant service
  - End-to-end encryption
  - Fine-grained access control

- **Google Cloud-native Integrations**
  - Cloud Functions for serverless event-driven computing
  - Dataflow (super powerful service in entire Google Cloud) for Stream Analytics
  - Cloud Logging

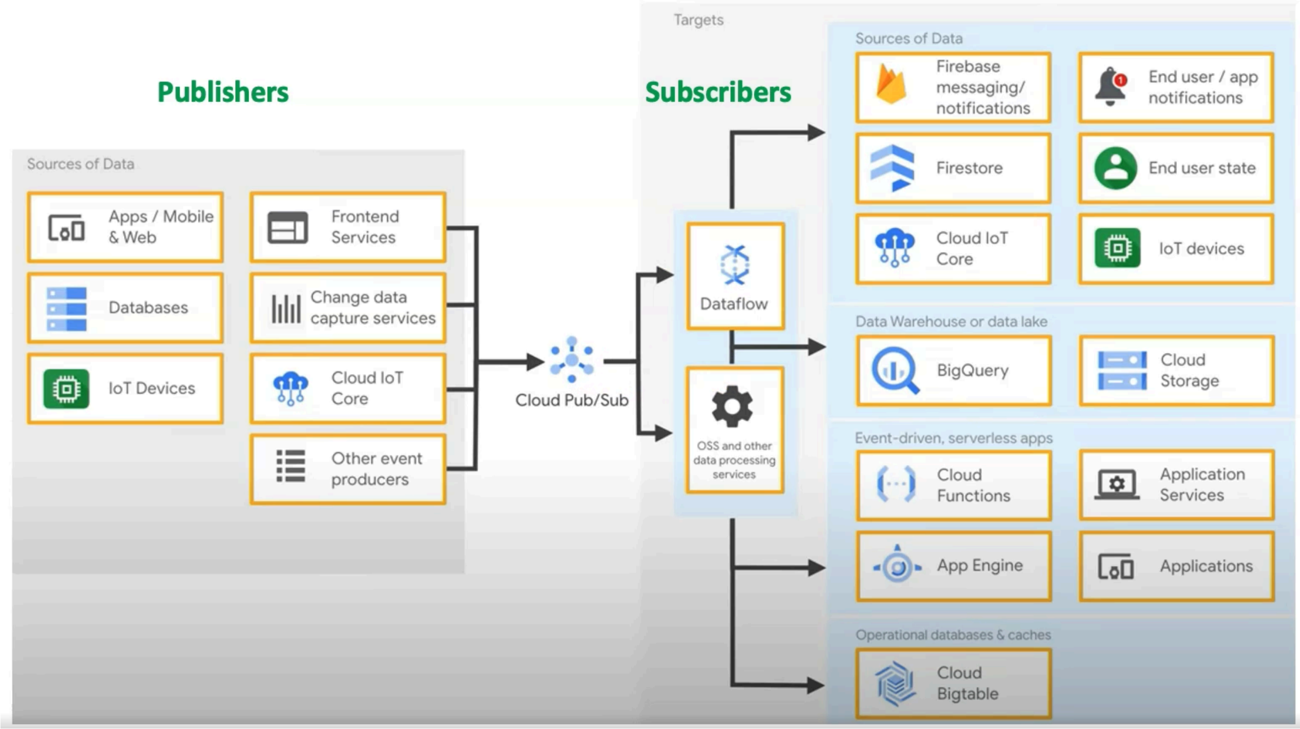
- **Message Filtering**
  - Subscribers will only receive messages that match the filter
  - Helps reduce delivery volume to subscribers

Google Cloud Pub/Sub

- **Dead Letter Topics**
  - To enable, need to create a Dead Letter Topic
  - Messages unable to be processed are published to Dead Letter Topic for later review and troubleshooting
  - This ensures that other messages aren't held up while issues are addressed.

- **Exactly Once Delivery**
  - Messages sent to the subscription are guaranteed not to be resent before the message's acknowledgement deadline expires

- Acknowledged messages will not be resent to the subscription


Publishers
Subscribers
Targets
Sources of Data
Apps / Mobile & Web
Frontend Services
Databases
Change data capture services
IoT Devices
Cloud IoT Core
Other event producers
Cloud Pub/Sub
Dataflow
OSS and other data processing services
Sources of Data
Firebase messaging/ notifications
End user / app notifications
Firestore
End user state
Cloud IoT Core
IoT devices
Data Warehouse or data lake
BigQuery
Cloud Storage
Event-driven, serverless apps
Cloud Functions
Application Services
App Engine
Applications
Operational databases & caches
Cloud Bigtable