This document provides information about how to create, delete, and administer Pub/Sub topics and subscriptions. For more information about publishing and accessing message data, see the Publisher Guide (/pubsub/docs/publisher) and Subscriber Overview (/pubsub/docs/subscriber).

You can create, delete, and view topics using the API, the Google Cloud Console, or the `gcloud` command-line tool. See the `gcloud pubsub` reference (/sdk/gcloud/reference/pubsub/) for a complete list of Pub/Sub API `gcloud` commands.

- For an introduction to the Cloud Console, see the Cloud Console Quickstart (/pubsub/docs/quickstart-console).
- When you create a topic, you can also specify your own encryption keys. See Using customer-managed encryption keys (/pubsub/docs/cmek).

You must first create a topic before you can publish or subscribe to it. Here is an example showing how to create a topic:

Here is an example showing how to delete a topic:

When you delete a topic, its subscriptions are not deleted, and the subscription's message backlog is available for subscribers. After a topic is deleted, its subscriptions have the topic name `_deleted-topic_`. If you try to create a topic with the same name as a topic you had just deleted, expect an error for a brief period of time after the deletion.

Here is an example showing how to get a list of topics:

By default, a maximum of 100 results are returned per query. You can specify an alternative value up to 1,000 using the page size parameter.

This section covers how to manage push and pull subscriptions. See the Subscriber Overview (/pubsub/docs/subscriber) for an overview and comparison of pull and push subscriptions.

You must create a subscription to a topic before subscribers can receive messages published to the topic.

Also see the gcloud reference (/sdk/gcloud/reference/pubsub/subscriptions) for subscriber commands.

Subscriptions have several properties that can be set at creation time or updated later, including:

- Delivery method: By default, Pub/Sub subscriptions use the pull method. You can switch to push delivery by specifying a non-empty, valid HTTPS push endpoint URL.You can switch back to pull delivery by specifying an empty URL.
- An acknowledgment deadline (/pubsub/docs/push#ack_deadline): If your code doesn't acknowledge (/pubsub/docs/push#receive_push) the message before the deadline, the message is sent again. The default is 10 seconds. The maximum custom deadline you can specify is 600 seconds (10 minutes).

You can use the following methods to create subscriptions:

- The Cloud Console:
  - If you know the topic ID, you can use the **subscriptions** section (https://console.cloud.google.com/cloudpubsub/subscriptions) of the console.
  - To create a subscription directly on the topic, go to the **topics** section (https://console.cloud.google.com/cloudpubsub/topics) of the console and open the topic.
- The gcloud pubsub subscriptions create (/pubsub/docs/quickstart-cli) command.
- The Client libraries:

You can create push subscriptions (/pubsub/docs/push) by providing an HTTP endpoint URL, as shown in these examples:

You can use the Pub/Sub Cloud Console (https://console.cloud.google.com/cloudpubsub/subscriptions) to change the subscription type. In the console, edit the subscription and change the **Delivery Type** to *Pull* or *Push*.

Alternately, you can use the gcloud command line tool (/pubsub/docs/quickstart-cli) command.. Specifically, use modify-push-config (/sdk/gcloud/reference/pubsub/subscriptions/modify-push-config) to modify the push endpoint URL.

- To convert a push subscription to pull, change the URL to an empty string.
- To convert a pull subscription to push, set a valid URL.

If you are switching from a push to a pull subscription, you should allow a short period of time for the change to take
The interval is typically on the order of a few seconds, but it can take several minutes in rare cases. During that time,
n may continue to deliver messages to the endpoint and you can expect that pull requests will generate errors.

Once a subscription has been created, you can update the push endpoint of a subscription or switch from push to pull and vice versa. Here are examples showing how to update an existing pull or push subscription by specifying a non-empty, valid HTTPS push endpoint URL:

By default, subscriptions expire after 31 days of inactivity. If Pub/Sub detects subscriber activity (such as open connections, active pulls, or successful pushes), the subscription deletion clock restarts.

To change this behavior, modify the subscription's expiration policy. You can set the policy duration when you create the subscription or modify it later. The minimum allowed value for an expiration policy is 1 day. There is no maximum. Use any of the following options to configure the policy:

- The duration and expiration flags available in the gcloud subscription (/sdk/gcloud/reference/pubsub/subscriptions/) commands. For example:

- Support for expiration policies is available in the Pub/Sub Client Libraries (/pubsub/docs/reference/libraries).
- RPC methods:
  - CreateSubscription (/pubsub/docs/reference/rpc/google.pubsub.v1#google.pubsub.v1.Subscriber.CreateSubscription)
  - UpdateSubscription (/pubsub/docs/reference/rpc/google.pubsub.v1#google.pubsub.v1.Subscriber.UpdateSubscription)
- REST method:
  - projects.subscriptions.create (/pubsub/docs/reference/rest/v1/projects.subscriptions/create)

By default, Pub/Sub ensures that subscriptions retain *unacknowledged* messages for 7 days from the moment of publication. You can set a subscription's `messageRetentionDuration` property to specify how long after publication messages should be retained, up to a maximum of 7 days. After the duration specified in this property, Pub/Sub is free to discard the message, regardless of its acknowledgment state. You can also configure the subscription to retain *acknowledged* messages for this duration. See Replaying and Discarding Messages (/pubsub/docs/replay-overview).

When you modify either the message retention duration or subscription expiration policy, the expiration duration mus
a value greater than the message retention duration. The defaults are 7 and 31 days, respectively.

You can list all subscriptions in a project, or only subscriptions to a particular topic. See Access Control (/pubsub/docs/access_control) for details on limiting access, if needed. Here are examples showing how to list all subscriptions in a project.

Here are examples showing how to access subscriptions for a specific topic.

Here are examples showing how to delete subscriptions in a project once they are no longer required.

A Pub/Sub resource name uniquely identifies a Pub/Sub resource, such as a subscription or topic, and must fit the following format:

`projects/project-identifier/collection/relative-name`

The *project-identifier* must be the project ID, available from the Google Cloud Console. For example, `projects/myproject/topics/mytopic`.

The *collection* must be one of `subscriptions` or `topics`.

The *relative-name* must:

- Not begin with the string `goog`.
- Start with a letter
- Contain between 3 and 255 characters
- Contain only the following characters:
  - Letters: `[A-Za-z]`

- Numbers: `[0-9]`
- Dashes: `-`
- Underscores: `_`
- Periods: `.`
- Tildes: `~`
- Plus signs: `+`
- Percent signs: `%`

  The special characters in the above list can be used in resource names without URL-encoding. However, you must ensure that any other special characters are properly encoded/decoded when used in URLs. For example, `mi-tópico` is an invalid *relative-name*. However, `mi-t%C3%B3pico` is valid.

  This is particularly important for REST (/pubsub/docs/reference/service_apis_overview) calls.

The client libraries (/pubsub/docs/reference/libraries) can take different approaches to naming resources, such as ying the *project* and the *relative-name* in separate fields.