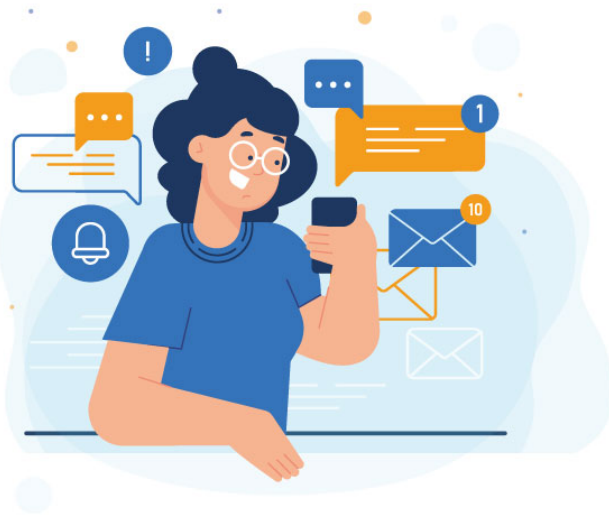


Pub/Sub 是什麼？6 大常見名詞與訊息生命週期介紹

Cloud Ace



Pub/Sub 是甚麼？

6 大常見名詞與訊息生命週期介紹

文章段落



- [Pub/Sub 是什麼？](#)
 - [概述與優勢介紹](#)
 - [6 大常見名詞](#)
 - [訊息生命週期](#)
- [ack / Retry 說明](#)
 - [ack](#)
 - [Retry](#)
- [常見問題](#)
- [結論](#)

Google Cloud 的 Pub/Sub 可為不同的應用程式之間提供收發訊息的功能，以及每則訊息的接收狀態追蹤功能，確保任何規模的訊息皆可穩定地傳送，有效提升用戶服務的靈活與穩健性，這次就讓我們從零開始，完整解析 Pub/Sub 功能優勢、6 大常見名詞與訊息生命週期吧！

Pub/Sub 是什麼？

概述與優勢介紹

Pub/Sub 是一種全球性訊息傳遞的服務，並具有高可用架構，可以在應用程式與服務間交換訊息。由 Publish 與 Subscribe 組成，運用了發佈及訂閱的概念讓訊息透過非同步的方式進行傳輸，只要您的服務可以處理訊息，Pub/Sub 就會持續傳送訊息，如果您 Subscription 下的所有 Subscriber 均無法處理訊息，Pub/Sub 會自動將您的訊息在默認的情況下保留 7 天（該天數最高可以延長至 31 天）。

另外在 Pub/Sub 中提供 3 種類型分別為：**多對一**、**多對多**、**一對多**，這 3 種類型可以讓使用者更靈活的處理訊息，讓服務可以高效運作。

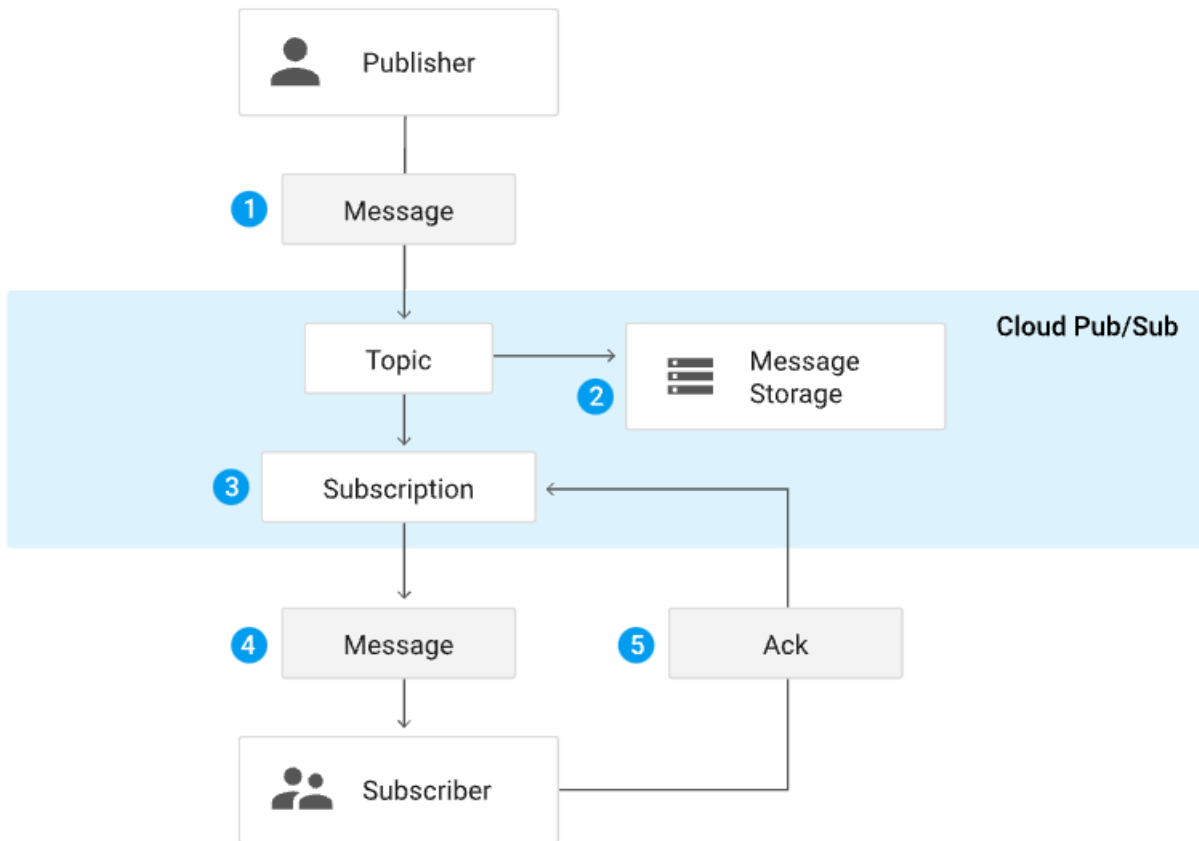
6 大常見名詞

對 Pub/Sub 服務有初步了解之後就要針對 Pub/Sub 的服務進行更深入的探討，首先我們先來了解 Pub/Sub 的六個常見名詞及生命週期：

- Publisher（發布者）：創建訊息並將訊息傳送到指定主題上的服務/人。
- Subscription（訂閱）：為 Subscriber 的上游，主要負責將訊息傳給指定的 Subscriber。
- Subscriber（訂閱者）：訂閱特定主題的服務/人。
- Topic（主題）：處理 Publisher 傳送的訊息並將其儲存至 Google Cloud Storage（簡稱 GCS）。
- Acknowledged（已確認）：簡稱「ack」主要用於確認訊息是否有被接收最少一次。
- Retry（重試）：若訊息處理失敗會進行 Retry 用來確保所有訊息都可以被處理的機制。

訊息生命週期

訊息生命週期流程（1-5）說明：（1）將訊息傳到 Pub / Sub，（2）將訊息寫入儲存空間，（3）Pub / Sub 將訊息發布給該主題的所有訂閱，（4）訂閱將訊息發送到訂閱者，（5）訂閱者回覆「ack」確認已經收到此訊息，並將此訊息從 GCS 上刪除；至此生命週期結束。



截圖自：[Cloud Pub/Sub 產品頁](#)

©2023 Google

可以發現特別是在流程（5）需要回覆「ack」給 Pub / Sub，這是因為 Pub / Sub 的基礎架構為「保證訂閱」的方式設計，確保所有訊息可以被傳送一次，因此為了確保所有資料完整被 Subscriber 接收到，會特別要求 Subscriber 進行「ack」，若沒有即時進行「ack」您的訊息則會被丟入「Retry」中。

ack / Retry 說明

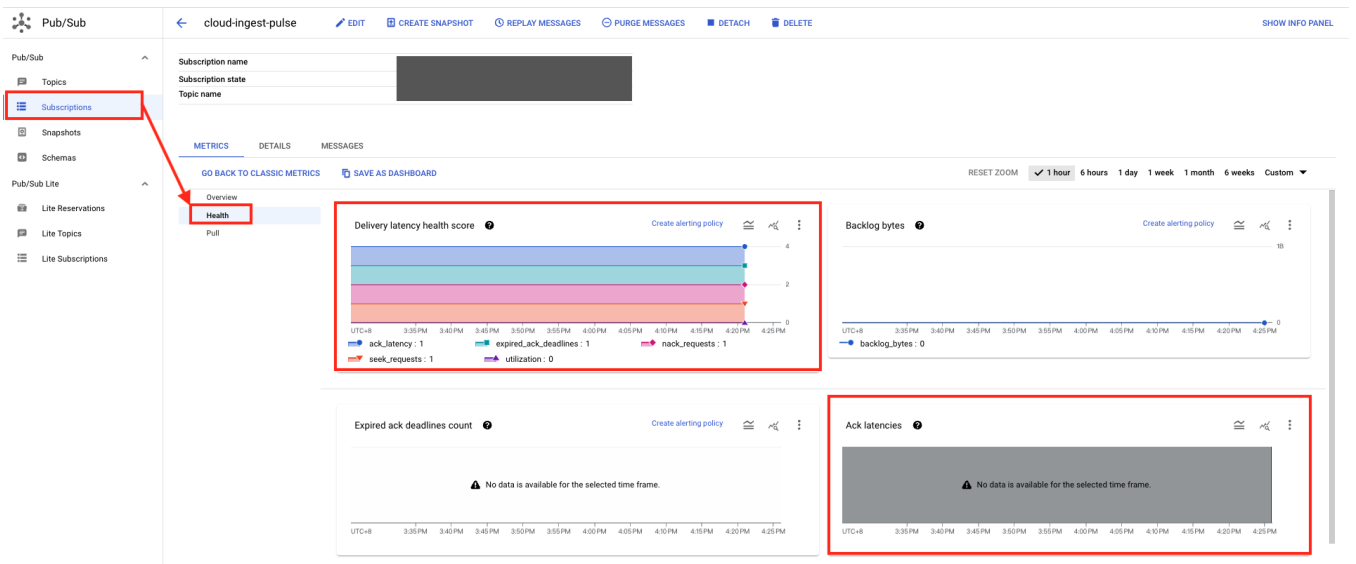
ack

若訊息未 ack 有何影響

若訊息未 ack 在 Pub / Sub 的設計上就會進行 Retry requests 的操作，這個動作會將未在時間內「ack」的訊息重新排序，這可能會導致您的 Subscriber 無法即時找到相關的訊息，進而導致該訊息延遲的狀況發生。

如何確認「未 ack」的訊息

可以透過「[subscription/delivery_latency_health_score](#)」、「[subscription/ack_latencies](#)」這兩個 Metric 來確認，其中「[subscription/delivery_latency_health_score](#)」若有數值為「0」代表該 Topic 過去 10 分鐘內可能有發生未正常處理的情況，比如：太晚進行「ack」、過去 10 分鐘內有 0.1% 的確認延遲時間超過 30 秒 ... 等；另外也可以從下圖右下角的「[subscription/ack_latencies](#)」看出延遲的時間。



截圖自：[Cloud Pub/Sub 產品頁](#)

© 2023 Google

Retry

甚麼原因會出現 Retry？

Retry 是因為 Subscriber 未在時間內進行「ack」，這些未被「ack」的訊息會被 Pub / Sub 視為 Subscriber [無法處理訊息](#)，且基於 Pub / Sub 為保證訂閱的服務，因此會將這類的訊息進行 Retry；若所有 Subscriber 均無法處理訊息，Pub/Sub 會將訊息保存至 GCS 中。

3 方法避免訊息被 Retry

如果服務處理速度跟不上 Pub / Sub 傳送的速度您可以透過以下方式進行修改：

- 提升服務的機器規格，讓您的服務可以即時處理 Pub / Sub 傳送的訊息。
- 可以透過延長「ack」時間來延後訊息被 Pub / Sub 視為無法處理。
- 可以加入[指數退避](#)的方式讓訊息在第一次失敗後等待最短指數退避的時間並重新傳送。

以上方法都可以盡量避免訊息被 Pub / Sub Retry，可以依照需求選擇最適合的方式。

常見問題

若訊息被 Retry 如何減緩影響？

如果您的訊息是擁有順序性的話，您可以為在[訊息加入排序](#)，可以讓來不及處理的訊息依照第一次的順序進行排序，可以避免您的服務請求順序出問題。

如何延長 ack 時間？

預設的「ack」時間為 10 秒，若要延長可以透過修改[modifyAckDeadline](#) 來延長「ack」的時間，最高可將「ack」時間延長至 60 分鐘。

若特定時間出現龐大流量該怎麼應對？

在 Pub / Sub 中提供使用[Flow Control](#) 的方式針對瞬時流量進行處理。

結論

在設計服務流程時訊息處理是相當重要的一環，傳統地端可能需要自行架設、維護 Kafka、RabbitMQ... 這類型的服務費時又麻煩，不過 Google Cloud 提供 Pub/Sub 服務可以使用，並結合了保證訂閱、多對多、延時處理... 等優點，讓使用者可以減少管理訊息相關的時間，更專注在開發、優化服務上。

這次介紹了 Pub/Sub 功能優勢，6 大常見名詞與訊息生命週期。各位若有進一步的 Pub/Sub 操作疑問，歡迎[聯絡 Cloud Ace](#) 獲得更進一步的資訊。

🕒 修訂版本 #2

★ 由 treeman 建立於 11 🕒 2025 18:07:23

✎ 由 treeman 更新於 11 🕒 2025 18:09:09