

<https://g.co/gemini/share/b3fc31d607e6>

好的，這兩個都是非常好的延伸提問，讓我們來深入探討這兩種「部分變化」的情境。

情境一：如果只有 **server-A** 修復了問題

這個情境非常關鍵，它展示了 Alertmanager 如何處理部分解決 (**Partial Resolution**) 的情況。

原始時間線回顧：

- **10:00:30** - 通知1: [FIRING:1] asd alert (instance: server-A)
- **10:05:30** - 通知2: [FIRING:2] asd alert (instances: server-A, server-B)

新的變化：

1. **10:15:00 - (關鍵點)** server-A 的問題被修復，但 server-B 仍然在報警。
 - Alertmanager 偵測到警報群組的狀態發生了變化（一個成員從 FIRING 變成了 RESOLVED）。
 - 因為群組內仍然有活躍的警報 (server-B)，所以不會發送 [RESOLVED] 的結案通知。
 - 相反，這被視為一次群組狀態的更新，因此會觸發 group_interval 的檢查。
2. **檢查 group_interval:**
 - 上次的通知時間是 10:05:30。
 - group_interval 是 5 分鐘，所以下一次最早的通知時間點是 10:10:30。
 - 現在的時間是 10:15:00，已經超過了 10:10:30。這意味著 Alertmanager 不需要再等待，可以立即發送更新。
3. **10:15:00 (立即) - 第三封通知 (FIRING 更新):**
 - 通知內容: [FIRING:1] asd alert (instance: server-B)
 - 這封通知的內容非常重要：它告訴你，現在只剩下 1 個警報在持續 (FIRING:1)，並且明確指出是 server-B。它同時也隱含了 server-A 已經恢復正常的資訊。
4. **接下來會發生什麼？**
 - 現在群組的狀態再次變得穩定（只剩下 server-B 一個成員）。
 - 如果 server-B 的問題一直沒有解決，也沒有其他新的伺服器加入，那麼系統就會切換到 repeat_interval 的邏輯。
 - 下一次的通知將會是 4 小時後，也就是在 **14:15:00** 左右，你會收到一封關於 server-B 的重複提醒。

一句話總結：

部分解決會觸發一次即時的「災情更新報告」，而不是「結案通知」。剩下的警報會繼續觸發它自己的重複提醒計時器。

情境二：如果問題在 11:26:17 被修復

這個問題很好，它在測試 `repeat_interval` 的計時器是否精確。答案是肯定的。

原始時間線回顧：

- **10:00:30** - 通知1: [FIRING:1] asd alert (instance: server-A)。
- 同時，一個 4 小時的 `repeat_interval` 計時器啟動，預計在 **14:00:30** 觸發。

新的變化：

1. **11:26:17 - (關鍵點)** server-A 的問題被修復。Prometheus 發送 RESOLVED 狀態。
2. 邏輯判斷：
 - `repeat_interval` 計時器的目標時間是 14:00:30。
 - 當前的時間 11:26:17 早於目標時間。
 - 因此，`repeat_interval` 的條件被中斷。那個預計在 14:00:30 響起的「提醒鬧鐘」被立刻取消了。
3. **11:26:17 (立即) - 第二封通知 (RESOLVED):**
 - 通知內容: [RESOLVED] asd alert (instance: server-A)
 - 事件在此刻正式「結案」。

結論：

這個結果與之前 11:30:00 修復的例子完全相同。無論是在 4 小時內的第 1 分鐘、第 90 分鐘、還是第 239 分鐘解決問題，結果都一樣：`repeat_interval` 計時器會被取消，並立即發送 RESOLVED 通知。

一句話總結：

只要在「提醒鬧鐘」響起前的任何一秒解決了問題，鬧鐘就會被關掉，並且你會立刻收到結案通知。