

Grafana Prometheus 設定指南



你的圖片主要是在整理 Grafana 與 Prometheus 的「抓取間隔 (scrape interval)」、「min step」、「interval」等概念，這些設定對於圖表正確顯示和 Alert 告警都很重要。以下幫你釐清幾個重點問題與名詞的差異與建議設定：

✅ 一、Grafana 的 `min step` 是什麼？

- `min step` 是 Grafana 向 Prometheus 查詢時的「最小資料間隔」，會影響查詢返回的資料點密度。
- 假如 `min step = 1m`，代表每 1 分鐘抓一筆資料（不會比這更細）。

“🔧 建議：`min step` 不應小於 Prometheus 的 *scrape interval*，不然會出現「no data」或看起來沒變化。”

✅ 二、Prometheus 的 `scrape_interval` 是什麼？

- 這是 Prometheus 多少時間去抓一次資料（例如 15s、30s、1m）。

- 寫在 `prometheus.yml` 裡的 job 裡，例如：

```
scrape_interval: 15s
```

“🔧 如果 Prometheus 是每 `15s` 抓一次資料，那麼 Grafana 的 `min step` 最小就可以設 `15s`，也可以設定大一點來減少負載，例如 `30s` 或 `1m`。”

✅ 三、Grafana datasource 的 `Min step` 要設多少？

- 若 Prometheus 的 `scrape_interval` 是 `15s`，則：
 - Grafana datasource 的 **Min step** 建議設為 **15s** 或以上。
 - 若設太小（如 `1s`），會抓不到資料，因為 Prometheus 並沒有這麼細的資料。
 - 設太大（如 `4m`）會讓曲線太平，看不到細節。

“✅ 你的問題「要設 `4m` 還是 `1m`」→ 建議設 **1m** 比較安全，若有特別需要再調小（前提是 Prometheus 有更細的抓取頻率）。”

✅ 四、Grafana 的 `$__interval` 是什麼？

- `$__interval` 是 Grafana 自動根據你瀏覽時間範圍和 panel 寬度來算出每個資料點間隔。
- 會被用在像 `rate()` 這類函數裡，讓時間區間可以自動變動。
- 可以被替換為像 `15s`、`1m` 這種時間格式。

✅ 五、`$__rate_interval` 又是什麼？

- ``$__rate_interval`` 是 Grafana 專門為 ``rate()``、``irate()`` 設計的變數，用來避免你寫死一個 interval。
- 通常你會看到這樣的查詢：

```
rate(http_requests_total[$__rate_interval])
```

✅ 六、`Min step` VS `Min time interval` 差別？

- ``Min step``：是 Grafana 送出查詢時的資料點間距（resolution）。
- ``Min time interval``：是 dashboard 或 panel 可允許的最小時間選擇範圍。
 - 比如你不希望使用者拉到 1s，這時候可以設 ``Min time interval = 15s``。

✅ 七、圖上的黃色線是什麼？

黃色垂直線在 Grafana 中通常代表：

1. **Alert 事件觸發時間點**
2. **Annotation（註解）事件**：可能你或某個 Alert Rule 插入的。

✅ 八、每個 panel 如何自定義查詢間距？

在 panel → Query 的最上方，會有一個 ``Resolution``：

- 可以選擇：
 - ``Default`` → 用全域 ``min step``

- ``1/2 step``、``1/10 step`` → 更細或更粗略
- 或直接在 PromQL 查詢內寫死 ``rate(...[1m])``

✅ 九、怎麼強制全部圖都用固定 ``minstep`` ？

在 Datasource 設定內：

- Global Min step 設成 1m
- 或在 dashboard variables 裡加入 ``$step = 1m``，然後在查詢裡用：

```
rate(metric_name[$step])
```

如果你能補充你現在的 ``prometheus.yml`` 的 ``scrape_interval`` 以及 Grafana 資料源的設定，我可以幫你確認是否配置正確。

要我幫你整理成一頁中英文筆記嗎？還是要我幫你圖解這些設定流程？



以繁體中文詳細說明:Prometheus scrape interval是1、grafana data source 設定是15秒，請問這樣我panel的rate()[時間要填多少]，15s、1m、還是4m



好的，我來以繁體中文詳細幫你說明這個情境：

問題背景整理：


- Prometheus 的 `scrape interval` = 1 分鐘 (60 秒)
- Grafana Data Source 的設定 = 15 秒 (min step)

你想問：

在 panel 查詢的 `rate()` 函數中，像這樣：

```
rate(cpu_usage_seconds_total[?])
```

這個中括號裡的時間區間 `[?]` 應該填 **15s**、**1m**、還是 **4m**？

 **正確答案：建議使用大於等於 2~3 倍 `scrape_interval` 的區間，也就是 `[2m]` 到 `[5m]` 之間。**

詳細說明：

1. `rate()` 是幹嘛的？

- `rate(metric[區間])` 是 PromQL 中用來計算「時間序列在這段區間內的平均變化速率」，常見於 counter 型 metric。
- 它會根據這段區間內的數值變化來估算速率。

2. 為什麼不建議填 `15s`？

- Prometheus 每 60 秒 (1 分鐘) 才 scrape 一次資料。
- 如果你用 `rate(metric[15s])`，這 15 秒區間內很可能只有 **1 筆資料或 0 筆資料**，Prometheus 根本沒東西可以計算 rate → 回傳 NaN 或 0，看起來圖是斷的。

3. 那填 `1m` 可以嗎？

- ``1m`` 剛好等於 scrape interval，只會有兩筆資料（最多），仍然不穩定。
- 通常 Prometheus 官方建議：**`rate()`** 的時間範圍要大於 **scrape_interval** 的 **2~4 倍**，這樣才能平均出比較穩定的變化趨勢。

4. 為什麼 ``4m`` 是常見選項？

- ``4m`` 能包含多筆資料點（約 4 筆），比較穩定。
- 適合 dashboard 顯示中長期趨勢，避免圖線震盪或斷掉。

實務建議：

<code>`scrape_interval`</code>	<code>`rate()`</code> 建議時間區間	說明
15 秒	<code>`[30s] ~ [2m]`</code>	很密，資料豐富，可用短區間
1 分鐘	<code>`[2m] ~ [5m]`</code>	建議區間！比較穩定
5 分鐘	<code>`[10m] ~ [15m]`</code>	要更長才能看到趨勢

你的情境（scrape = 1m）建議：

```
rate(cpu_usage_seconds_total[4m])
```

或

```
rate(cpu_usage_seconds_total[$__rate_interval])
```

搭配 ``Grafana`` 自動算的 ``$__rate_interval`` 可以更動態，根據時間範圍自動選擇合適的區間。

小技巧：

- 想看短期細節波動 → 用 ``irate()`` 搭配 ``[1m]``，但圖可能會抖動。
 - 想看平滑趨勢 → 用 ``rate()`` 搭配 ``[4m]`` 或 ``$__rate_interval``。
-

有需要我幫你寫一段範例 dashboard 查詢或圖表設定嗎？或你也可以貼出你目前用的 ``PromQL`` 查詢語法，我幫你優化一下。



如果Prometheus 的 `scrape interval = 1 分鐘 (60 秒)`、
Grafana Data Source 的設定 = 15 秒 (min step)，那
minstep要設多少



這是一個非常好的問題，我來幫你釐清 **Grafana** 的 ``min step``（或叫 **Min interval**）該怎麼設定，並解釋這個設定的影響：

基本定義回顧

名稱	說明
Prometheus scrape interval	Prometheus 幾秒抓一次資料（例如 <code>`1m`</code> ）

名稱	說明
Grafana min step (Min interval)	Grafana 查資料的最小間隔，防止太細的查詢打爆 Prometheus
Grafana query step	實際查詢時 Prometheus 被要求的資料解析度，由 Grafana 根據時間範圍計算出來

🎯 你的情境：

條件	值
Prometheus 的 `scrape_interval`	`1m`（每分鐘抓一次）
Grafana 的 `Min step`	目前設 `15s`

🚫 問題在哪？

- Prometheus 每 **60 秒** 才有一筆資料，如果你在 Grafana 設定 `Min step = 15s`，代表 Grafana 會想要「每 15 秒要抓一筆資料」，這會導致：

“👤 查不到資料，或只會抓到一樣的值，甚至出現空白/NaN。”

✅ 正確建議：Min step 設為「不小於 scrape_interval」

“📌 建議設定：`min step = 1m` 或更大，例如 `2m` 也可以”

✅ 範例設定方式：

在 Grafana 資料來源 (Data source) → Prometheus → `Min interval` 欄位，請設定為：

1m

🔍 為什麼這樣設比較穩？

1. 避免抓不到資料 (因為 Prometheus 根本沒那麼頻繁)
2. 減少查詢壓力 (太小間隔會導致 Prometheus 重複傳一樣的資料，浪費資源)
3. 讓圖表更平順、更準確

📌 補充說明：min step ≠ panel 裡的 query 間隔

Grafana 實際查詢時，會根據你選的時間範圍和圖寬度，自動算出一個 **step**，例如你選一張圖時間範圍是 6 小時、圖寬 300px，Grafana 可能會算出：

$\text{step} = 7200\text{秒} / 300\text{px} = 24\text{秒}$

這時候 Grafana 會用：

`http://prometheus/api/v1/query_range?step=24s`

這個 `step` 會自動與 **Min step** 做比較，取較大值：

```
step = max(自動計算值, min step)
```

✅ 總結（你的例子）：

設定項目	建議值
Prometheus scrape interval	`1m`
Grafana Data Source Min step	✅ 設為 `1m`（與 scrape interval 一樣）
Grafana Panel 內 `rate(metric[?])`	建議 `rate(metric[4m])`

需要我幫你畫一張圖解這些之間的關係嗎？還是你要我幫你整理一頁設定參考表？



Grafana Data Source 的設定 = 15 秒（min step），minstep設1m可以嗎



可以的，而且在你的情況下 `min step` 設為 `1m` 反而是更建議的設定。我來詳細說明原因：

✅ 你的情境整理：

項目	值
Prometheus `scrape_interval`	`1m` (每分鐘抓一次資料)
Grafana Data Source 的 `Min step`	目前是 `15s` ❌
你想問的	改成 `1m` 可以嗎？

✅ 正確解釋與建議

❌ 不建議設 `min step = 15s` 的原因：

- Prometheus 每 **60 秒** 才有一筆資料。
- 若 Grafana 設定 `min step = 15s`，但 Prometheus 根本沒這麼密，會導致：
 - 查詢回來都是「重複資料」或「NaN」。
 - 浪費 Grafana 查詢資源，圖表顯示也會不準。

✅ 建議：`min step = 1m` (或更大)

這樣 Grafana 至少每次查的資料點距離會大於等於 Prometheus 真正可提供的時間間距，才會：

- ✅ 拿到實際存在的資料點
- ✅ 不會出現資料點為空或跳動
- ✅ 減少不必要的 Prometheus 查詢負擔

📌 圖示觀念：

時間	資料是否存在？
00:00:00	✅ Prometheus 有資料
00:00:15	❌ 無資料
00:00:30	❌ 無資料
00:00:45	❌ 無資料
00:01:00	✅ Prometheus 有資料

設 `min step = 15s` 時，Grafana 會每 15 秒問一次 Prometheus，但 Prometheus 實際只有 1 分鐘才給你一次資料，中間全是空白。

✅ 設定範例（建議這樣做）：

在 Grafana → Data Sources → Prometheus 設定畫面中：

- 找到 `Min interval`
- 改成：

1m

✅ 其他建議

查詢功能	建議參數
<code>rate()</code> 區間	<code>rate(metric[4m])</code>

查詢功能	建議參數
`Min step`	`1m` (等於 scrape_interval)
`Min time interval` (選配)	例如設 `1m`，避免 dashboard 時間選太細

✅ 結論

“✅ 可以把 **Grafana** 的 `min step` 改為 `1m`，不但可以，還是正確做法！”

這樣能讓 Grafana 跟 Prometheus 的資料頻率對齊，避免抓不到資料或圖表異常。

若你需要我幫你寫一段對應的 PromQL 或 dashboard 查詢樣板，我也可以幫你補上～



以繁體中文詳細說明:Min step vs Min time interval 差別



當你在使用 **Grafana** 搭配 **Prometheus** 時，常會看到兩個容易混淆的設定項目：

“**Min step** 和 **Min time interval**”

它們雖然看起來類似，但其實用途與影響完全不同！



一句話區分：



名稱	功能
 Min step	限制「Grafana 查詢 Prometheus 的最小資料間距」
 Min time interval	限制「使用者在 Grafana 介面上可以選擇的最小時間範圍」

詳細比較

項目	Min step	Min time interval
用途	控制 Grafana 發出查詢時的「最小 step」值（資料解析度）	限制時間選擇器的「最短可選範圍」
設置位置	Grafana 資料來源（Data Source）設定畫面	Dashboard 或 Panel 的設定中
影響的對象	Prometheus 查詢的 resolution（每隔幾秒抓一筆）	使用者在 UI 上能選的時間範圍（如 1m、5m、15m）
常見設定值	`1m`, `30s`, `15s` 等等	`1m`, `5m`, `10m` 等等
錯誤設定會導致？	資料抓不到 / 圖表斷線 / 資料重複	使用者不能選短時間來看圖

一起看範例

Min step

- 假設 Prometheus 每 1 分鐘抓一次資料（scrape_interval = 1m）
- 如果你在 Grafana 資料來源裡設：

Min step = 15s ❌

那會造成：

- Grafana 每 15 秒查一次資料
- 但 Prometheus 實際上沒那麼多資料，圖會「斷斷續續」

✅ 正確做法是：

Min step = 1m (與 Prometheus 對齊)

◆ Min time interval

這是你希望使用者不要選太短的時間範圍時使用，例如防止負載過重：

假設你在 Grafana 的 Panel 設：

Min time interval = 5m

這樣使用者即使手動輸入 `1m`、`2m`，Grafana 也會強制改成 `5m`，防止查太細的資料拉太多 request。

🎯 實務總結：

場景	設定建議
Prometheus 每 1 分鐘抓一次資料	Min step 設 1m
不想讓使用者選太短時間（例如 10 秒）	Min time interval 設 1m 或 5m

場景

設定建議

要避免圖表破圖或空白

確保 ``Min step` ≥ scrape_interval``

兩個設定位置補充：

在資料來源（Data source）裡：

-  設定 ``Min step``

在 Dashboard 或 Panel → ``Time range``：

-  設定 ``Min time interval``

如果你想看圖解或設定畫面，我也可以幫你整理一張對照表或操作示意圖～是否需要我補一張圖幫你整理？



以繁體中文詳細說明: Grafana ``$__interval``、``$__rate_interval``、Min step的關係



這三個變數與設定常讓人混淆，分別是：

“``Grafana` 的 `$__interval`、`$__rate_interval`、以及資料來源的 `Min step`。”`

它們彼此有關聯，但用途不同，我來幫你一一說明並畫出它們的「關係圖」。

一句話總結

名稱	作用	誰決定它
`\$__interval`	Grafana 自動計算的查詢「step」	由 Grafana 根據「時間範圍」與「圖寬」算出
`\$__rate_interval`	給 `rate()` 函數用的區間，避免你自己寫死	基於 `\$__interval` 推算出來的
`Min step`（資料來源設定）	限制 `\$__interval` 的下限（不能太小）	你在 Data Source 設定時填的

Grafana 的查詢過程中會怎麼處理？

假設你選擇顯示過去 1 小時（3600 秒），Panel 寬度為 300px：

1 Grafana 自動算出：

```
$__interval = 3600秒 / 300px = 12s
```

但你在 Data Source 中設定了：

```
Min step = 30s
```

2 Grafana 檢查：

```
真正使用的查詢 step = max(12s, 30s) = 30s
```

3 所以最終：

- 查詢 step = **30 秒**
- ``$__interval`` = **30s**
- ``$__rate_interval`` = **30s 或更大** (通常 = $2 \sim 4 \times \$_interval$, 例如 1m、2m)

三者功能與差異詳細說明

名稱	用途	影響什麼
<code>`Min step`</code>	限制 Grafana 查詢 Prometheus 時的最小間隔	降低查詢頻率、防止打爆 Prometheus
<code>`\$__interval`</code>	自動計算查詢解析度的變數，用來當作 <code>`step`</code> 值	影響每筆資料點距離多久抓一次
<code>`\$__rate_interval`</code>	專門給 <code>`rate()`</code> / <code>`irate()`</code> 裡使用的時間範圍	讓 <code>`rate()`</code> 的區間隨時間自動調整

PromQL 範例

```
rate(cpu_usage_total[$__rate_interval])
```

在這裡 ``$__rate_interval`` 可能會變成：

- 你時間範圍很大時 → 變成 ``[5m]``、``[10m]``
- 你時間範圍很小時 → 變成 ``[30s]``、``[1m]``

這樣可以避免你寫死 `[1m]` 導致在大範圍時圖表不準確。


三者關係圖（邏輯順序）

 使用者選「時間範圍」 + Panel 寬度


↓

 Grafana 算出 $\$_interval = \text{總秒數} / \text{寬度}$

↓

 跟 Data Source 的 Min step 比較：若太小 → 強制套用 Min step

↓

 計算 PromQL 查詢解析度： $\text{step} = \$_interval$ (受 Min step 限制)

↓

 給 $\$_rate_interval$ 推算時間範圍給 `rate()`

小提醒：

錯誤做法

後果

「Min step」設太小（如 1s）但 Prometheus 每 1m 抓一次

抓不到資料、圖表 NaN

`rate()` 寫死 `[1m]` 卻看 24h

資料不足，曲線會亂跳

實務建議

設定	建議值
Prometheus scrape_interval	`15s`, `30s`, `1m` (依實際情況)
Grafana Data Source Min step	同上，設為不小於 scrape_interval
Panel 使用 ` \$__rate_interval `	✅ 建議！讓 rate() 自動適應範圍
避免在 rate() 寫死 `[1m]`	除非你非常確定要分析短期變化

如果你想要我幫你畫成一張完整圖解或整理成 PDF 笨人包筆記，也可以告訴我，我可以幫你製作～



以繁體中文詳細說明: panel的md、interval



當你在 Grafana 的每個 **Panel** 中設定查詢時，會遇到兩個常見的參數：

“✅ `interval`
 ✅ `intervalMs` (有時你會看到變數叫 `__interval_ms`)
 ✅ `max data points` (顯示為 `MD`，最大資料點數)”

這些設定會共同影響：

- ◆ Grafana 發送給 Prometheus 的查詢解析度 (step)
- ◆ 你圖上看到的資料點密度與滑順程度
- ◆ 查詢效能與資料負載

我來幫你詳細說明每個的意義與差異：

一、什麼是 `interval` ？

定義：

“`interval` 是 **Grafana** 計算出的查詢解析度 (*step*)，表示：
→ 每隔多久送出一筆查詢點（每隔幾秒抓一筆資料）。”

`interval` 的決定因素：

Grafana 會根據 **兩個參數**動態計算 `interval`：

參數	說明
 時間範圍 （例如過去 1h、6h）	時間範圍越長 → 資料點不能太多 → `interval` 變大
 Max data points (MD)	即「希望最多回傳幾筆資料」 → 整體資料點數受限

計算公式（概念）：

```
interval = time_range / max_data_points
```

例如：

- 如果你看的是 **1 小時 (3600 秒)**
- ``maxDataPoints = 300``
- 那麼：

```
interval = 3600 / 300 = 12 秒
```

二、什麼是 `intervalMs` ？

這只是 `interval` 的毫秒 (ms) 版本，也叫 `\$__interval_ms`，很少手動用到，除非你在查詢中有需要毫秒單位（通常在自訂資料源或 JavaScript 外掛中才會用到）。

三、什麼是 `Max data points (MD)` ？

定義：

“這是 **每張圖最多顯示幾筆資料點（折線上的點）**，設定在 *Panel → Query* 設定頁的下方。”

如果你的時間範圍是 6 小時，而 `Max data points = 100`，那就會自動把整個查詢切成 100 等分去抓資料。

預設值

- 預設是根據你的圖表寬度自動推算（例如寬度為 800px，預設 MD 可能就是 800）

MD 太小的影響？

- `interval` 會被拉大

- 圖表變粗略，看不到細節（每筆資料代表很多秒）

四、panel 的查詢中 `interval` 與 `\$__interval` 差異？

- `\$__interval` 是 Grafana 幫你算出來的「實際查詢步長」。
- `interval` 是你自己可以在 Variable 裡控制的預設單位，例如：

```
interval:
  type: interval
  name: step
  options: ["15s", "30s", "1m", "5m"]
  default: 1m
```

你可以用 `step` 這類變數控制 query 用的時間區間。

使用建議

項目	建議值
`Max data points` (MD)	`300 ~ 1000` 較為適中
時間範圍很大時	用 `\$__rate_interval`，自動適配時間長短
長時間圖表卡頓、查詢太慢	嘗試調高 `interval` 或調小 MD

實務 PromQL 寫法範例：

```
rate(http_requests_total[$__rate_interval])
```

或者用：

```
avg_over_time(memory_usage[$__interval])
```



整體關係圖總結

時間範圍 + MD (max data points)



計算出 `$__interval`



- 查詢時作為 `step`
- 可套用到 PromQL 裡



若 `interval` 太小 → 會被 `Min step` 強制上修

如你需要我幫你畫一張圖來圖解 `interval`、`min step`、`MD` 之間的邏輯，我可以直接生成圖表給你。有需要嗎？



以繁體中文詳細說明:要怎麼各panel 設定不同的時間範圍呢?



這是一個很實用也很常見的需求：



“ 我希望 **Grafana Dashboard** 上不同 **Panel** 可以有不同的時間範圍，而不是跟整頁 Dashboard 一樣。”

這在觀察某些長期趨勢（如 CPU 使用率）和短期細節（如最近 10 分鐘 log 波動）時非常重要。

Grafana 支援每個 Panel 使用「自訂時間範圍（Time Range override）」

只要簡單幾個步驟，就能讓每張圖使用自己的時間範圍，不受整個 Dashboard 控制。

📋 操作步驟教學（以繁體中文說明）：

步驟一：打開 Panel 編輯畫面

1. 點選你要設定的圖表（Panel）
2. 點右上角的「編輯」（Edit）

步驟二：設定 Panel 的時間範圍

1. 在編輯畫面中，點選上方的「Time range（時間範圍）」分頁
2. 勾選 `Override relative time` → 輸入你要的時間，例如：
 - `last 5m`：只看最近 5 分鐘
 - `1h`：最近 1 小時
 - `24h`：過去一天

3. （可選）勾選 `Add time shift`（時間偏移）：

- 可以用來比較「現在」和「昨天同一時間」
- 例如輸入 `1d` → 圖表會顯示「昨天的資料」

步驟三：儲存

- 點「應用（Apply）」或「儲存 Dashboard」
- 就完成設定，該 Panel 的時間範圍就會獨立於 Dashboard 的整體時間選擇

補充：常見時間格式

格式	說明
`5m`	過去 5 分鐘
`1h`	過去 1 小時
`24h`	過去一天
`7d`	過去一週
`now-1h`	從現在往前 1 小時
`now-2d/d`	昨天整天

實務應用情境舉例：

Panel	Override 時間設定	用途說明
面板 A	<code>`5m`</code>	觀察最近 CPU 使用率細節
面板 B	<code>`1h`</code>	看最近 1 小時的記憶體變化
面板 C	<code>`24h`</code>	長期分析 log 數量
面板 D	<code>`1h` + `time shift=1d`</code>	與昨天同一時間比較負載變化

小提醒：

- 使用 ``Override relative time`` 不會影響其他 Panel
- 時間格式和 Dashboard 選擇器相同，支援 ``now-xxx`` 語法
- 如果你有使用 ``$__interval``，它也會根據這個 Panel 的時間自動重新計算，完全獨立

如果你想我幫你畫一張 **圖解「全頁 vs Panel 個別時間範圍」差異** 或 **製作一頁中文教學備忘單 PDF**，也可以跟我說喔！



以繁體中文詳細說明:要怎麼各panel 的ministep 都設定成 1m，而不用分開設定呢？



你問的是一個非常實用的 **Grafana 優化技巧**：

“🔍 「我想讓所有 Panel 的查詢 min step（最小查詢間隔）都設為 `1m`，而不用每一張 panel 分開設，該怎麼做？」”

答案是：在 **Data Source**（資料來源）設定中統一設定 `Min interval` 即可。

✅ 什麼是 Min step (= Min interval) ？

在 Grafana 查詢 Prometheus 時，每張圖表都會根據時間範圍和寬度計算出查詢間隔（step），但這個值不能小於你在 **Data Source** 裡設定的「Min interval」。

這個 `Min interval` 就是你說的「每張 panel 的 minstep」。

📋 操作步驟（讓全部 Panel min step 都是 1m）：

1 開啟資料來源設定

1. 進入 Grafana 左側選單 → ⚙️「設定（Settings）」→ 點選「**Data sources**（資料來源）」
2. 找到你用的 Prometheus 資料來源，點進去

2 設定 Min interval

1. 找到 `**Min interval**` 欄位（有時顯示為 `Minimum resolution step`）
2. 輸入：

1m

意思是：「Grafana 查 Prometheus 時，每筆資料點最小間隔為 1 分鐘」

3. 儲存資料來源設定

3 完成 🎉

從現在開始：

- 所有 Panel 的 `\$__interval` 都會被限制為 **最小 1 分鐘**
- 你不需要每張 panel 設定 `Min step`
- 也不需要 panel 裡手動寫死 step，只要用 `\$__interval`、`\$__rate_interval` 都會自動生效

🔍 視覺化關係圖

Dashboard Panel 查詢時間範圍 + 顯示寬度

↓

Grafana 自動計算 interval

↓

與 Data Source 的 Min interval 比較

↓

最終查詢 step = max(計算值, Min interval)

✅ 實務範例

```
rate(http_requests_total[$__rate_interval])
```

即使你不寫死 step，只要 Min interval 設為 `1m`，Grafana 自動計算出來的 `\$__rate_interval` 也不會小於 1 分鐘，穩定又省資源。

進階補充：還可以這樣強化

- 若某些 Panel 真的需要不同 step，可以：
 - Panel → 編輯 → Query 區 → `Resolution` 改成 1/2、1/5 step
 - 或手動覆蓋 PromQL 的 `[interval]`

總結

目的	做法
所有 panel min step 都設為 `1m`	✅ 到 Data Source 設定 `Min interval = 1m`
不用每張 panel 手動改	✅ 因為 `\$__interval` 自動根據這個下限

如你想我幫你出一份「Prometheus + Grafana 資源最佳化設定筆記」或「中文設定懶人包 PDF」，也可以告訴我，我可以幫你整理製作！



以繁體中文詳細說明:grafana的alert功能、跟prometheus的alert 有什麼不同?



非常好的問題！Grafana 和 Prometheus 都有 **告警 (Alerting)** 功能，但用途、架構與運作方式不同。以下我會用繁體中文幫你清楚地比較與說明這兩者：

🔔 一、Grafana Alert vs Prometheus Alert 差異總覽

比較項目	Grafana Alert	Prometheus Alert (Alertmanager)
🇮🇹 基礎來源	Panel 查詢結果 (PromQL or 整合資料來源)	純 Prometheus 的時間序列資料
🧠 判斷邏輯	在 Grafana 中透過 UI 設定條件	在 Prometheus 的 `alerting rules` 中以 YAML 寫死
🕒 評估頻率	設定每 X 秒評估一次 (UI 設定)	`evaluation_interval` 設定
📧 通知機制	Grafana Alert → 通知通道	Prometheus → Alertmanager → 通知通道
🔧 建立方式	Grafana 介面上設定 (內建 GUI)	`prometheus.yml` + `alert_rules.yml`
💡 優勢	易於視覺化、支援多資料來源 (如 Loki, Elasticsearch)	深度整合 Prometheus 生態、靈活轉發與靜默管理
🧩 整合	可支援非 Prometheus (如 Loki、InfluxDB)	只能使用 Prometheus 指標
📦 狀態管理	可顯示於 Dashboard，並支援警報群組 (Group)	Alertmanager 可自動去重、分派、分組、靜默處理
🔗 接收器整合	Email、Slack、LINE Notify、Webhook、Opsgenie 等	同上，透過 Alertmanager 管理

二、Grafana Alert 功能詳解

Grafana 自從 v8 開始推出 **統一告警系統 (Unified Alerting)**，具備以下功能：

特點：

- 支援多種資料來源：**Prometheus、Loki、InfluxDB、Graphite...**
- UI 可視化操作，適合「視覺化告警條件」
- 告警狀態（OK、Pending、Alerting）可顯示於圖表旁
- 支援 Alert Rule、Notification Policy、Contact Point（完整告警流程）

典型應用範例：

“如果某個 Panel 顯示 CPU > 80% 持續 5 分鐘 → 發 Slack 通知”

三、Prometheus Alert 功能詳解

Prometheus 的 Alert 是透過 `.yaml` 文字檔定義的 rule，搭配 Alertmanager 處理通知流程。

特點：

- 告警條件寫在 `prometheus.yaml` 或獨立 `alert_rules.yaml`
- 支援複雜的 PromQL 表達式（像 `rate()`，`avg_over_time()`）
- 靠 **Alertmanager** 處理告警轉發、靜默、分組、通知整合

- 經常與 `Thanos`、`VictoriaMetrics`、`Cortex` 等搭配大型監控架構使用

典型設定範例：





```
groups:
  - name: example_alerts
    rules:
      - alert: HighCPUUsage
        expr: avg(rate(cpu_usage_total[5m])) > 0.8
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "CPU 使用率過高"
```

四、兩者何時該選哪個？

使用情境	建議用法
想快速在圖表上加告警、非程式人員使用	✅ 用 Grafana Alert
有完整的 DevOps 架構、需統一管理通知	✅ 用 Prometheus + Alertmanager
想用非 Prometheus 的資料來源（Loki, Elasticsearch）告警	✅ 用 Grafana Alert
需要 Alert 靜默（Silence）、分派通知、告警分組等複雜流程	✅ 用 Alertmanager

整體建議（實務架構搭配）

最佳實務搭配方式：

1.  用 Prometheus 撈數據、撰寫 Alert Rule
2.  用 Alertmanager 處理告警轉發與管理
3.  用 Grafana 顯示告警狀態 + 視覺化圖表
4.  額外需求（非 Prometheus 資料來源）可用 Grafana Alert 補充

← 結論

功能	Grafana Alert	Prometheus Alert
易用性	★★★★★	★★
可視化	★★★★★	★
可擴展性（靜默、重試、群組）	★★	★★★★★
統一監控架構整合	★★	★★★★★

如果你有特定用途（例如：要對接 LINE Notify、Email、Opsgenie）、要整合 Slack，或者想看怎麼把這兩種 Alert 混合運用，我也可以幫你出一份實務設計範例。需要嗎？



以繁體中文詳細說明:為什麼panel的某個時間點會出現黃色縱向虛線呢?



在 **Grafana** 的圖表 (Panel) 中，如果你發現某個時間點有一條：

“ **黃色縱向虛線 (垂直虛線)**」出現在圖上”

那這並不是圖表錯誤，而是 Grafana 有意提供的一種視覺提示。下面我會用繁體中文幫你詳細說明這是什麼、什麼時候會出現、該如何設定或關閉。

黃色縱向虛線是什麼？

這條黃色虛線是 **Annotation (註解)** 或 **Alert 事件觸發點**，代表 在那個時間點發生了某個事件。

主要可能的兩種情況：

① 【常見】這是「Alert 事件觸發時間點」

- 如果該 Panel 上有設定 Grafana 的告警 (Alert rule)，當條件被觸發時，Grafana 會自動在圖上加一條：

“ 黃色虛線 (告警觸發的時間點)”

其他特徵：

- 滑鼠移過去會顯示告警名稱與觸發時間
- 可配合告警狀態切換 (OK → Alerting → OK)

② 【也可能是】使用者或系統「Annotation 註解事件」

- Grafana 允許你將特定時間點加上「註解 (annotation)」
- 例如：部署紀錄、自動化指令執行、運維事件記錄...
- Annotation 來源可以是：
 - 手動新增（右鍵選時間 → 加註解）
 - 自動從資料來源來（ex: 監控系統、CI/CD 工具）

Annotation 特徵：

- 也會顯示在圖上為黃色虛線
- 可以設定是否要套用到所有 Panel 或單一 Panel

如何檢查是哪種原因？

方法一：滑鼠移到黃色線上方

- 如果是 **Alert**，會顯示 Alert 名稱、狀態變化時間
- 如果是 **Annotation**，會顯示「Annotation 名稱」、「Tag」、「描述內容」

方法二：開啟 Panel → 編輯 → 左側點「Annotations」

你會看到是否有 Annotation 層級被啟用：

- 系統預設可能有：

Alerts → 來自告警的事件

Annotations → 使用者自定義事件

- 可以點選或取消選擇來控制要不要顯示在圖上

Bonus：如何關閉黃色虛線？

方法一：關閉某個 Panel 的 Alert 顯示

- Panel → 編輯 → 左側「Annotations」
- 把 `Alerts` 取消勾選即可

方法二：關閉全部 Annotation（全頁面）

- Dashboard → 設定 → Annotations → 把不要的來源刪掉或關閉

常見實務用途

黃色虛線來源	用途範例
Grafana Alert	告警觸發時間（如 CPU > 80%）
CI/CD Annotation	紀錄某次部署、重啟時間點
手動註記	運維工程師在某次異常後手動加註說明
外部來源	由 API 自動推送，例如 Jenkins、GitHub Actions 發布資訊

結論

現象	原因
Panel 上出現黃色縱向虛線	可能是告警事件（Alert）或註解事件（Annotation）
滑鼠移上去會出現提示框	可辨別是 Alert 還是註解
可自行設定要不要顯示	Panel 編輯 → Annotations 設定頁可管理

如果你願意貼出畫面截圖，我可以幫你判斷那條線是哪一種來源。如果你有設定自動 Annotation（如部署紀錄），我也可以幫你寫個自動紀錄範例。需要嗎？