

06 · 05 · 2025

為什麼你裝了一堆 O11y 工具  
卻沒人用？



Observability ≠ Shopping + Installing



DevOpsDays Taipei 2025



**Hello! I'm smalltown**

**MaiCoin Group 打雜小弟大叔**

**主要涉略範圍: SRE, IT, Data, QA**

**時常擔任團隊首位推動 DevOps 的人**

**目前 SRE 團隊 6 人 ↑**

**努力讓瑣事維持在 60% ↓**

大家所維運的服務 24/7 不打烊  
停機就等於燒錢！



可觀測性失敗的真正成本 = 停機時間 x 每分鐘損失

?

想說安裝完 Grafana 和 Prometheus 就搞定了



但是...工具≠能力

!

不過遇到問題仍然是... ssh + tail -f



用肉眼在每秒幾萬行的 Log 瀑布找問題

!

# 多到爆炸的指標！但找不到真正的問題



收集 10 萬條 metrics, 找不到轉換率掉 7% 的原因

!

或是會看到 Dev 跟 Ops 互踢皮球



!

## 誤區一：以工具為中心的導入方式



安裝 → Default Dashboard → 下班

!

## 誤區二：只有 Ops 在乎指標



指標會說話，只是除了 Ops，沒人聽得懂它在說什麼

!

## 誤區三：噪音警報 ≠ 可觀測性



每天早上醒來, Slack 上的警報比我的未讀訊息還多



先把提問寫進程式，再讓答案寫進圖表

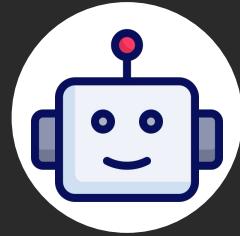


# 從三個面向讓 O11y 在團隊內落地



## 文化與角色

讓 Dev、Ops 與管理層共  
享責任，形成推動可觀測  
性的基石



## 流程與自動化

確保開發與運維一體化，  
不再依賴手動追蹤



## 平台與治理

讓平台可持續運作，避免  
工具閒置與浪費



# 共同訂出對齊業務的 SLO/SLA



**Business KPI**

「為什麼要做可觀測性」的根本驅動



**Service Level**

讓團隊有清晰可以監控與追蹤的 SLO/SLA



**Telemetry**

提供實時與歷史的參考，驗證服務是否符合期望

從業務痛點出發，定義可觀測性需求



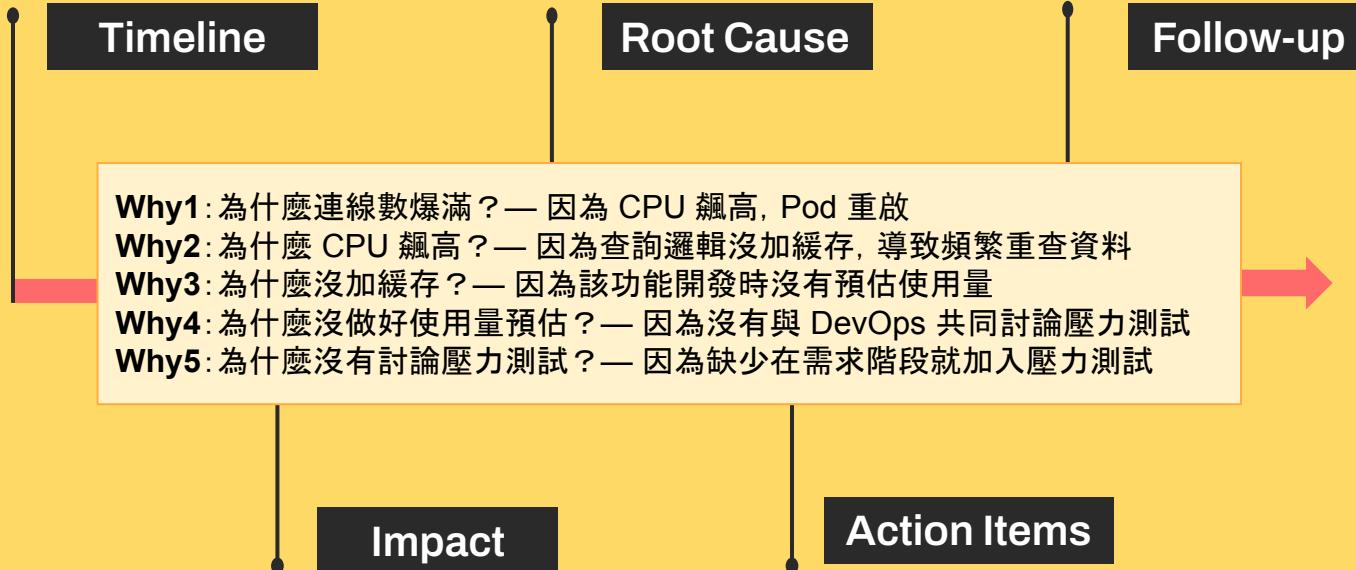
# 除了 Ops, Dev 也要一起輪值

	MON	TUE	WED	THU	FRI	SAT
Ops	Noah	Lucas	James	Michael	Jack	Daniel
Dev	William	David	John	Cooper	Adam	Andrew

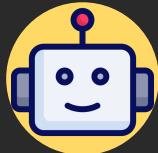
Ops: 拒當人肉路由器



# 在 Incident Review 中學習，而不是抓戰犯

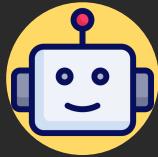


用「5 Whys」而非「誰的錯」來檢討問題



先挑一條最賺錢／最常出事的路徑做 E2E 追蹤





# 在應用服務的生命週期中埋藏線索



Build

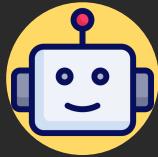
Git SHA, Version...etc



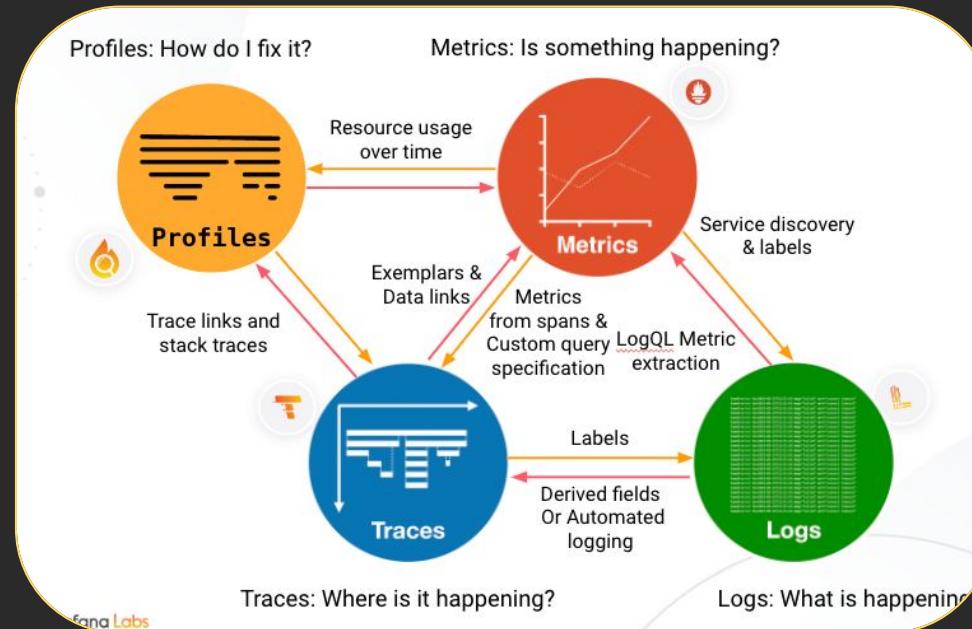
Execution

Trace ID, Span ID, Log,  
Metric...etc

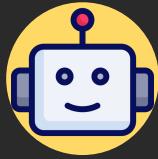
假如什麼都沒有的話，今年的通靈王大賽就開始了！



# 將 Observability 3+1 本柱關聯起來



Alert → Metrics → Trace → Log → Trace → ...



# Runbook 有沒有不存在的一天 → RaC

Document

XXX as Code

Service

再拖一陣子就可以都交給 AI Agent 就好了



# Hot／Warm／Cold Data 各有其用



**Hot Data**

近 7 天放 SSD  
即時告警使用



**Warm Data**

30 天放 HDD  
事故回溯使用



**Cold Data**

180 天上 Object Storage  
稽核合規使用

Cold Data 要定期檢視是否正常，不然...



# 成本治理三步驟

1

刪除無用 Metrics  
降低高雜訊 Alert

2

設定 Budget Alert  
自動化通知

3

建立 RACI 矩陣  
定期審核

透過 RACI (Responsible, Accountable, Consulted, Informed) 矩陣

明確各項監控與採購行為中「誰負責」和「誰付錢」



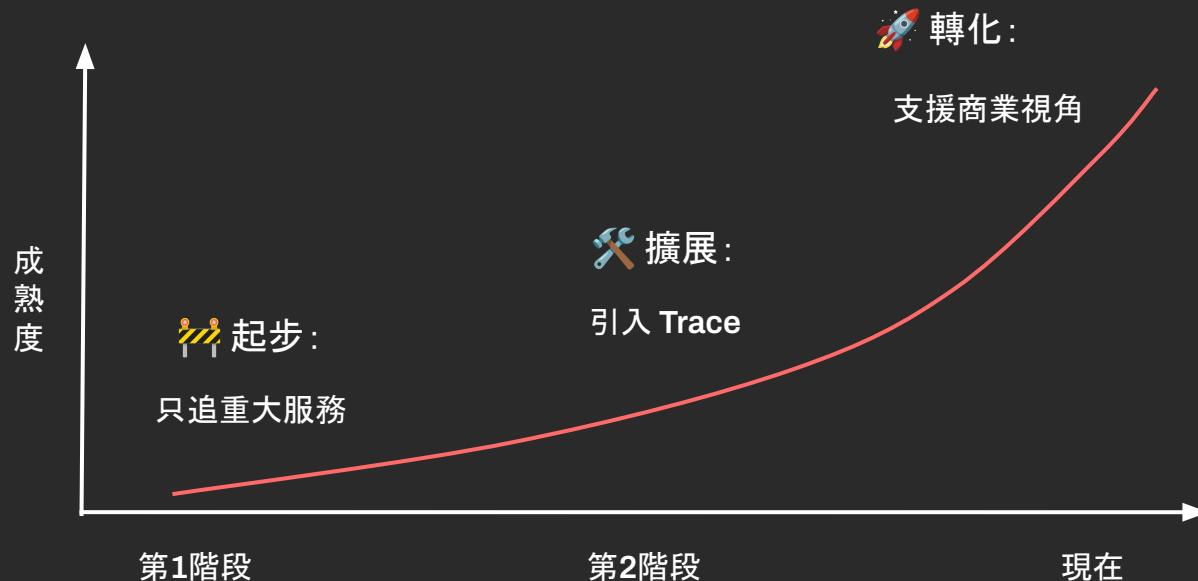
# 設計儀表板時，先問「這個指標能幫助誰？」



不要讓 Dashboard 只說工程師聽得懂的話



# 沒有最好，只有更好



以前追 Log 是靠第六感，現在靠 Trace，未來靠 AI。（希望啦）

工具是手段  
問題定義才是王道





# ！培養見人說人話，見鬼說鬼話的能力

Tech Signal	Dev Focus	Ops Focus	Manager Focus	Conclusion
P95 Latency ↑	查程式/SQL	主機負載	客戶體驗下降	我們的結帳流程本週慢了300ms, 可能影響轉換率
Alert Noise 過多	Debug 困難	OnCall 頻率上升	士氣降低、加班成本	降 60% 噪音 = 少兩次夜班 加班費



# 套用 P-I-M-A 的說話框架



## Problem

用非技術或業務語言  
描述目前的異常現象



## Impact

量化這個問題對業務  
或系統造成的直接損  
失或風險



## Metric

指出唯一可驗證或反  
駁此影響的技術指標  
或數據



## Action

下一步應立刻執行的  
技術或流程措施

P: 小明把程式裡「Debug 全開」，每天狂丟「INFO」、「DEBUG」到 CloudWatch Log

I: 結果 CloudWatch Log 量一夕暴增，老闆月底看到帳單愣住，以為 AWS 帳號被盜

M: CloudWatch Log Insights 顯示當日寫入日誌大小超過 5 TB

A: 立刻把 Log Level 調回 Error → 改用 FilterPattern 過濾關鍵資訊 → 設定每日 Log 用量告警

# 1

列全公司 Top 3  
停機痛點

# 2

選 1 條關鍵路徑  
開始追蹤



下週就能做  
的幾件小事

# 3

在 Build 流程加  
入 Git SHA,  
Version

# 4

清掉 3 個無人使  
用的 Dashboard

# Thanks!



Do you have any  
questions?

~~不問問題的話，今天晚上電話會響喔~~





# We're Hiring!

一起來當同事吧！

- [Senior Site Reliability Engineer](#)
- [Senior Data Engineer](#)
- [Senior IT Engineer](#)
- [Blockchain Engineer \(Wallet Team\)](#)
- [\(Senior\) Backend Engineer](#)
- [Micro Service Software Engineer](#)
- [Cyber Security Engineer](#)