

從零到一：搭建你的第一個 Observability 平台

DevOpsDays Taipei 2025

劉義瑋 Blueswen

自我介绍



- 劉義瑋 Blueswen
- [blueswen](#) @ GitHub
- DevOps Engineer @ Berry AI
- 領域
 - DevOps
 - Observability
 - Developer Experience

Outline

- 10 min - Observability 基礎概念介紹
- 20 min - Observability Signals - Data Flow 與交互應用
- 10 min - Hands-on：Observability 平台
- 10 min - Alerting 介紹與實作
- 10 min - Hands-on：Alerting
- Recap



圖片來源：[Grafana](https://grafana.com/)

目標

- 使用 **Docker Compose** 建立 **Observability 平台**，釐清資料流與各元件功能
- 了解可觀測性資訊 **Logs**、**Metrics**、**Traces** 的生成、收集、儲存、使用
- 學習在 **Grafana** 中**交互應用** Logs、Metrics、Traces 資訊，並設定**告警規則**

Dev or Ops ?
Python or Java ?

你是怎麼排查問題的呢？



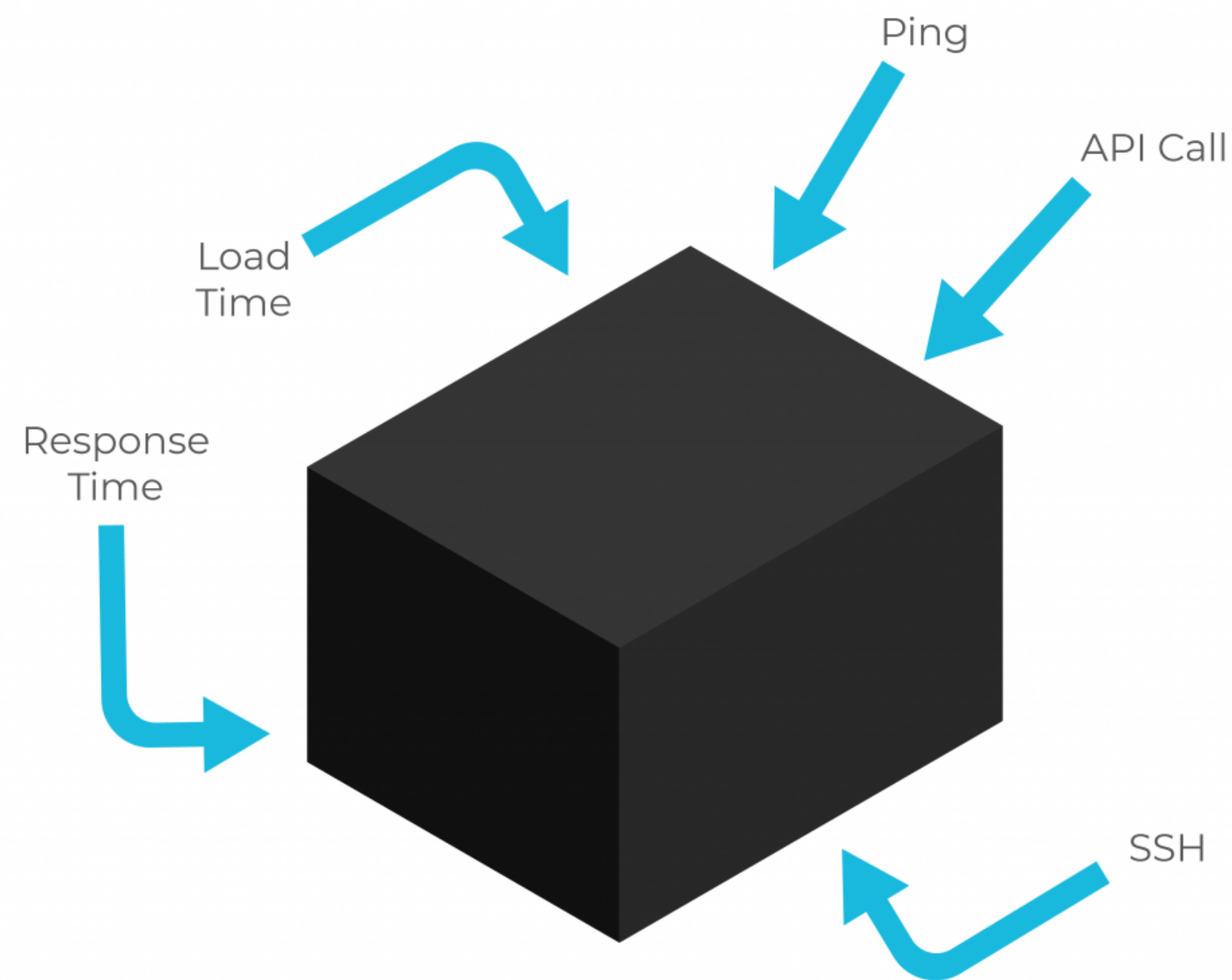
你也是被冠上天才之名的通靈奇才嗎？

強化 Observability 可觀測性，人人都能通靈

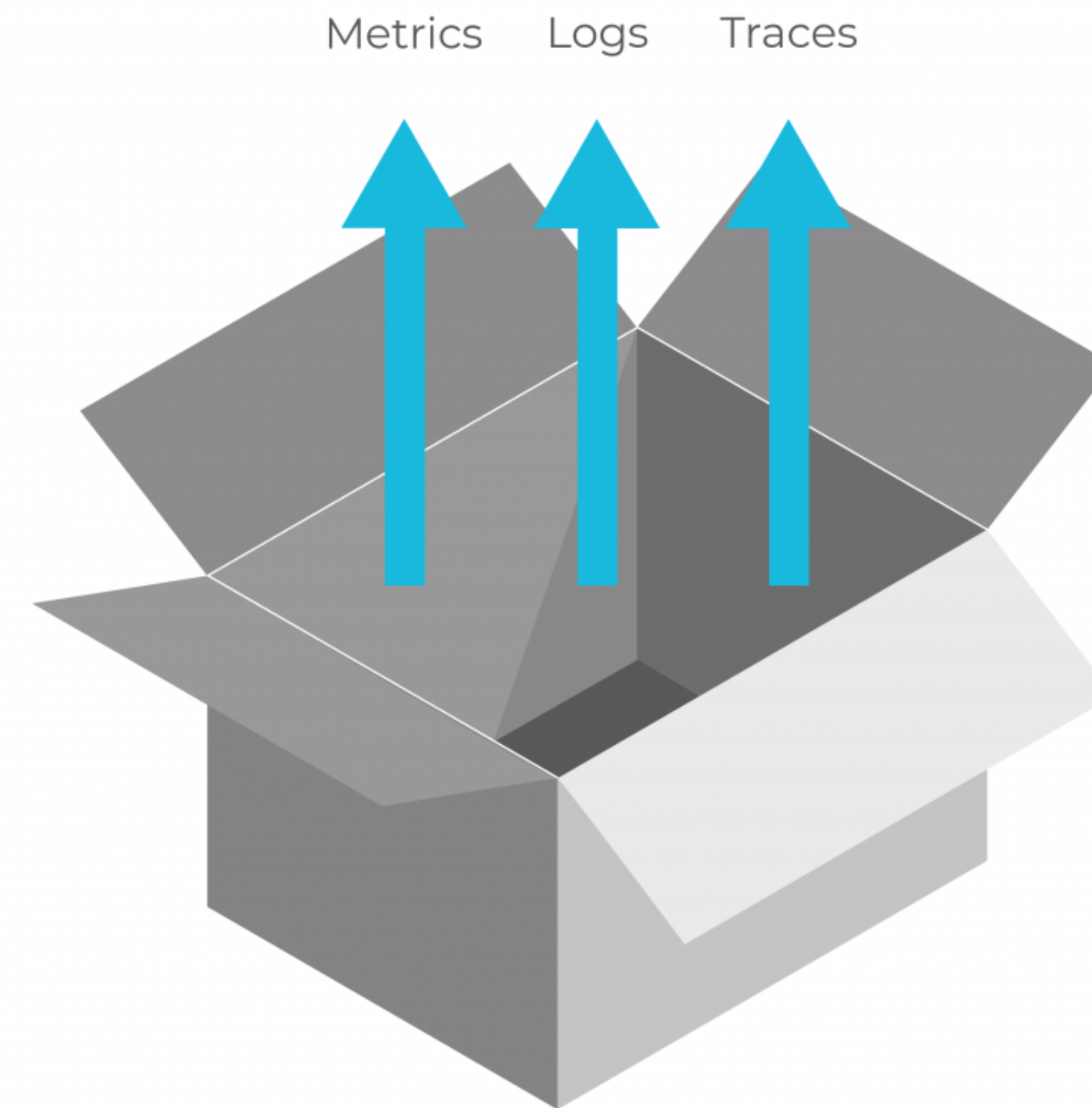
Observability 基礎概念介紹

Observability Intro

系統可被觀測或測量的一種特性



Blackbox Monitoring



Whitebox Monitoring

Observability

圖片來源：[Multi-Cloud Monitoring: A Cloud Security Essential](#)

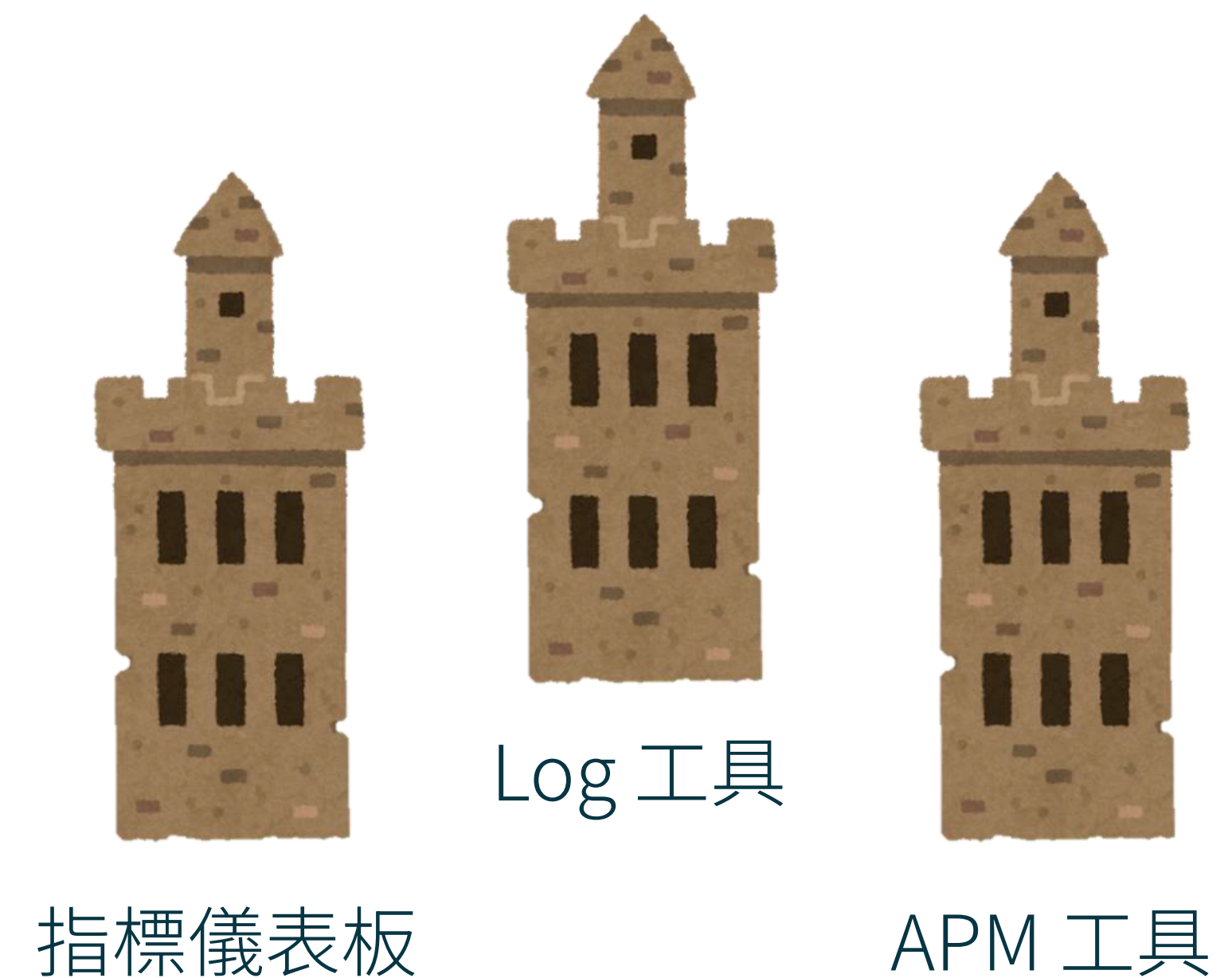
Observability Intro

透過各種資訊，清楚了解系統狀態

有足夠的資訊嗎？



是否散落各處，形成 Data Silo？



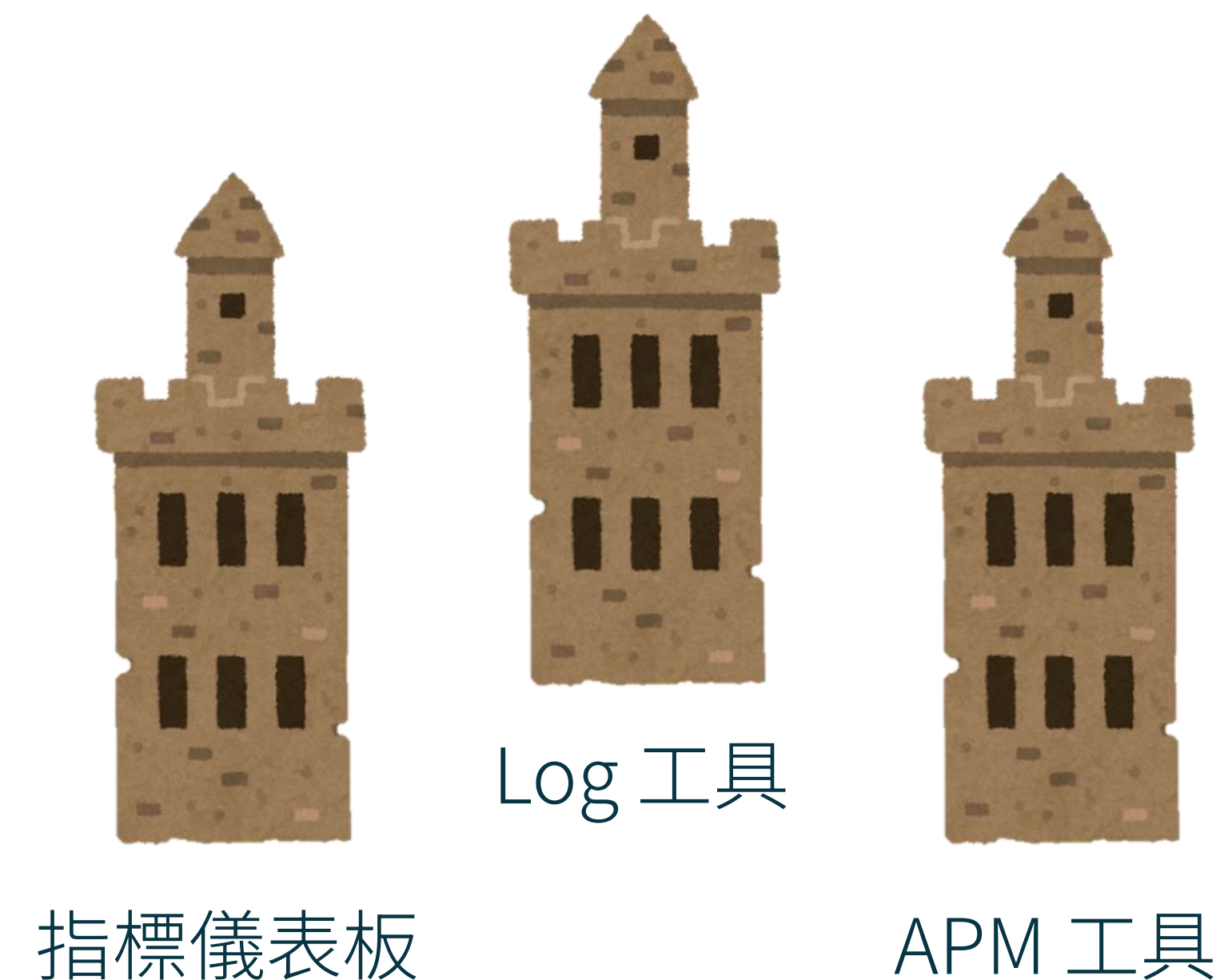
Observability Intro - Signals

透過各種資訊，清楚了解系統狀態

有足夠的資訊嗎？



是否散落各處，形成 Data Silo？



Observability Signals - 可觀測性資訊

系統**主動揭露**讓我們能夠**更理解**它的資訊



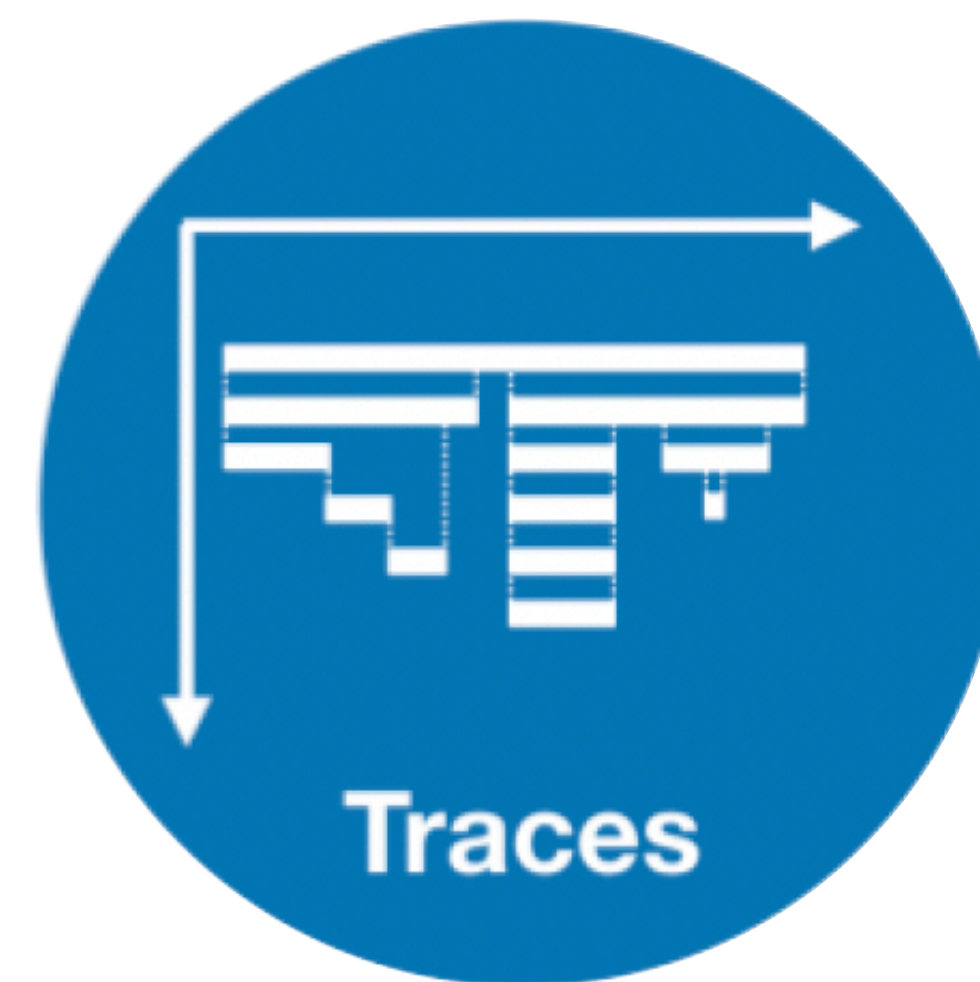
指標

不同時間採樣的系統量化指標
如：CPU 使用率、API 回應時間



日誌

紀錄系統中發生的事情
如：Debug 訊息、Exception



分散式追蹤

紀錄行為在不同服務中的歷程
如：SSO 行為橫跨多個服務

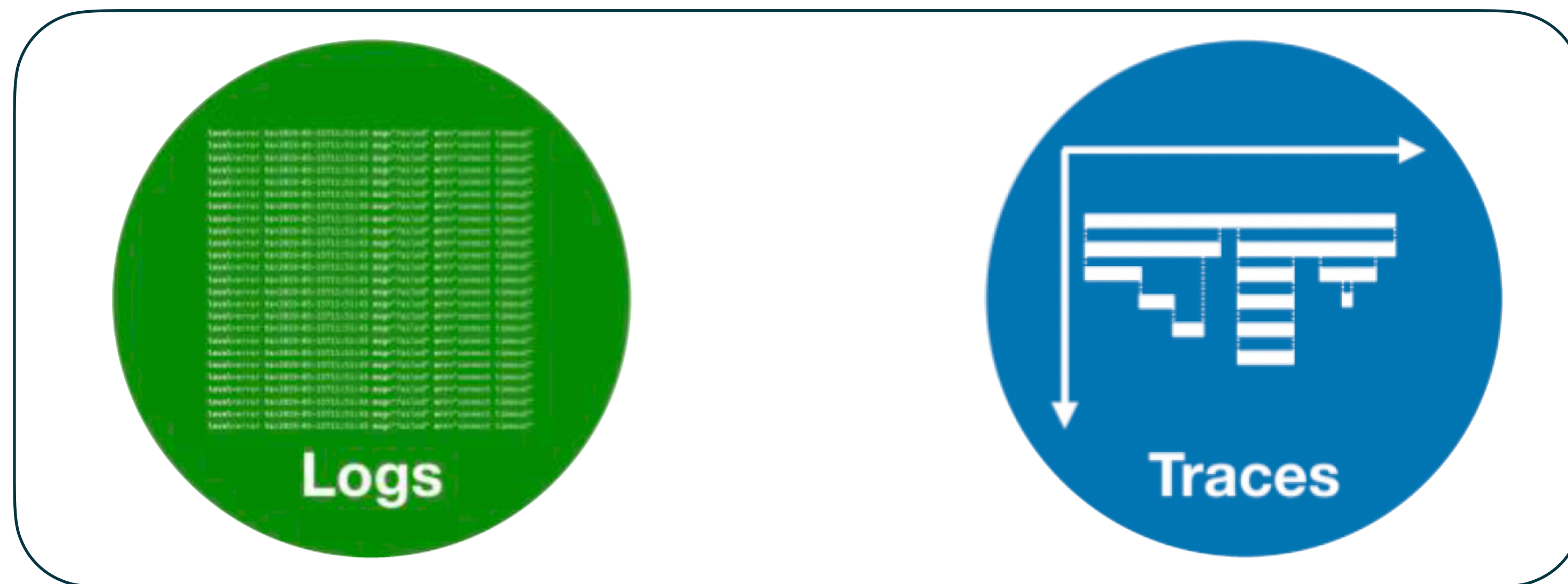
Observability Signals - 可觀測性資訊

系統**主動揭露**讓我們能夠**更理解它**的資訊



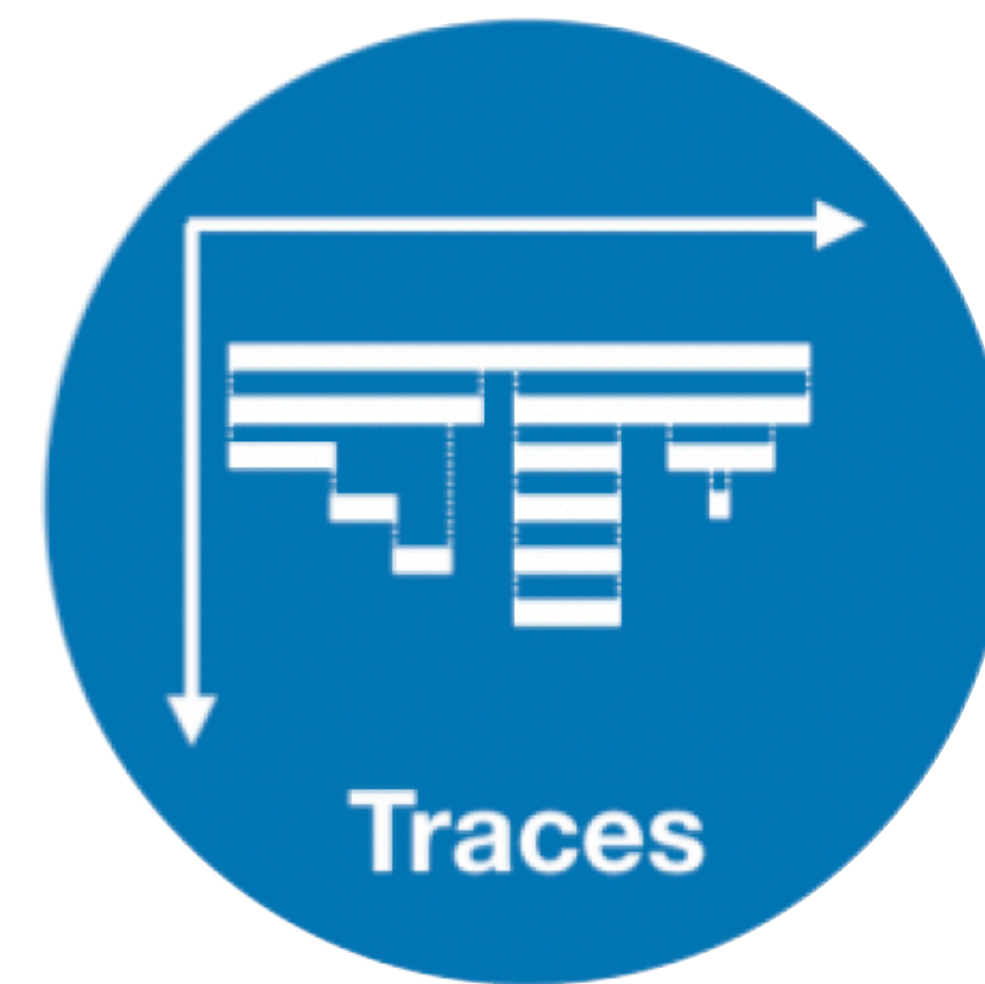
狀況如何

徵狀：有問題發生



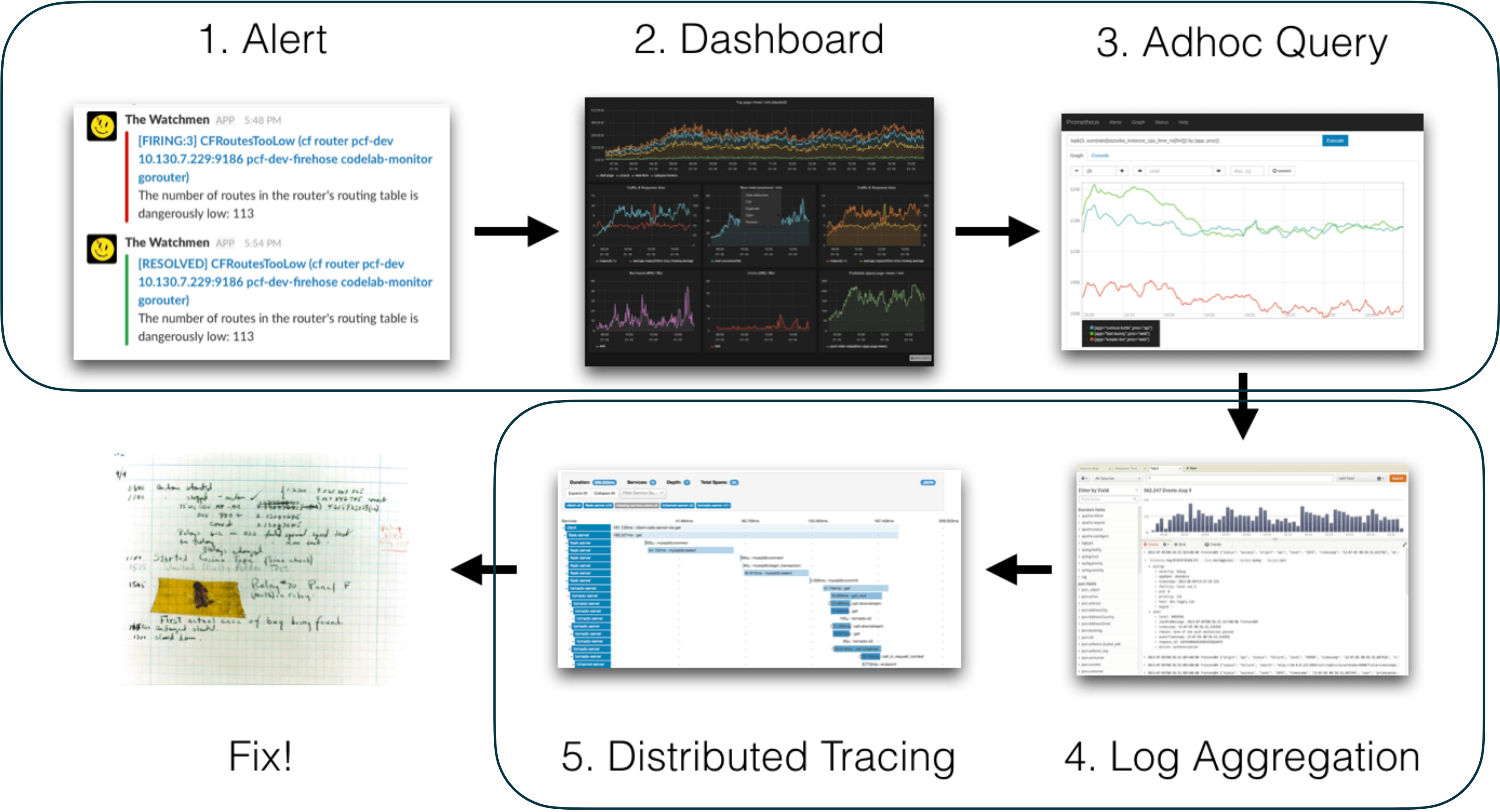
發生什麼事

脈絡：怎麼發生的



哪裡發生的

Observability Signals - 應用於問題排除



徵狀

脈絡

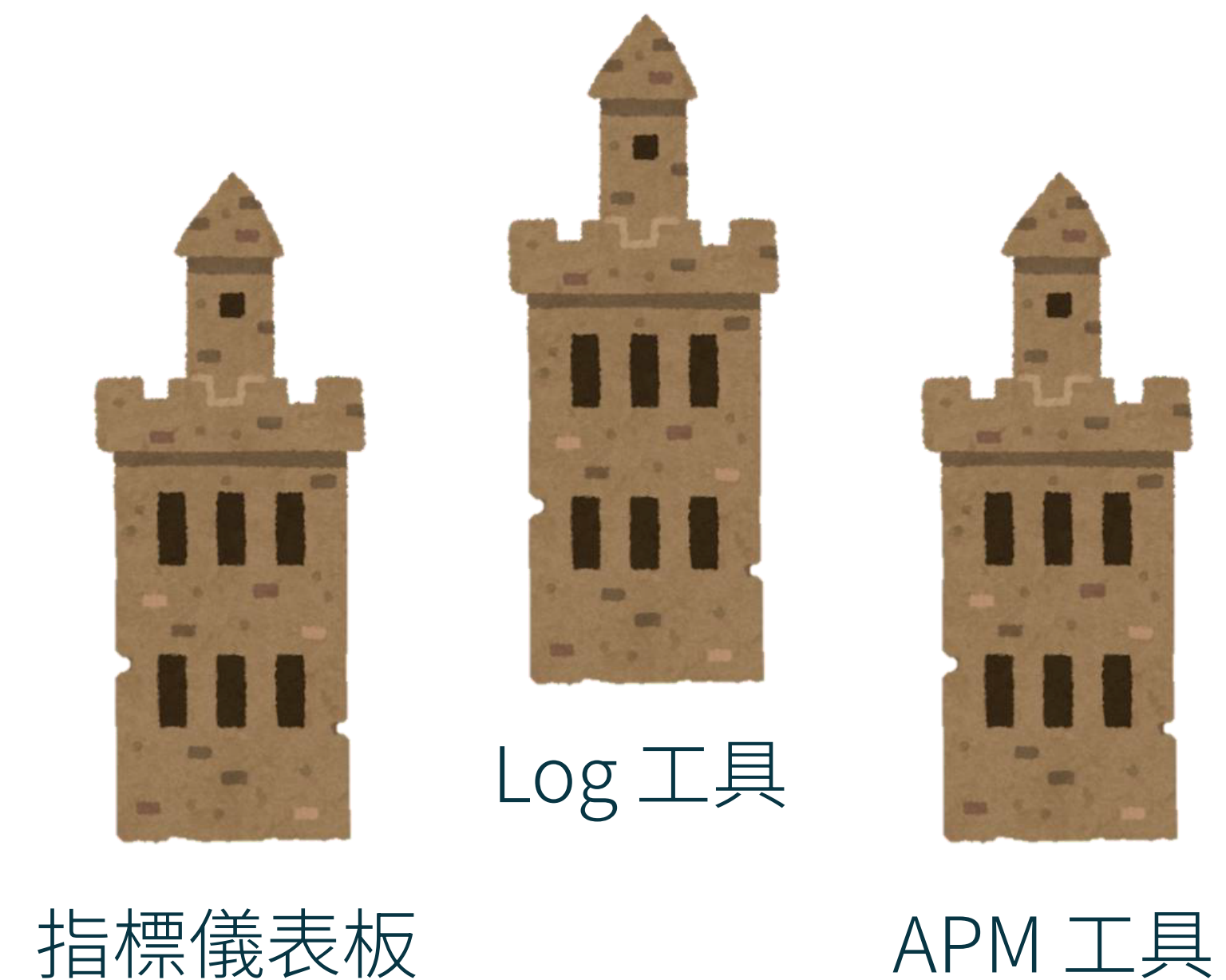
Observability Intro - Platform

透過各種資訊，清楚了解系統狀態

有足夠的資訊嗎？



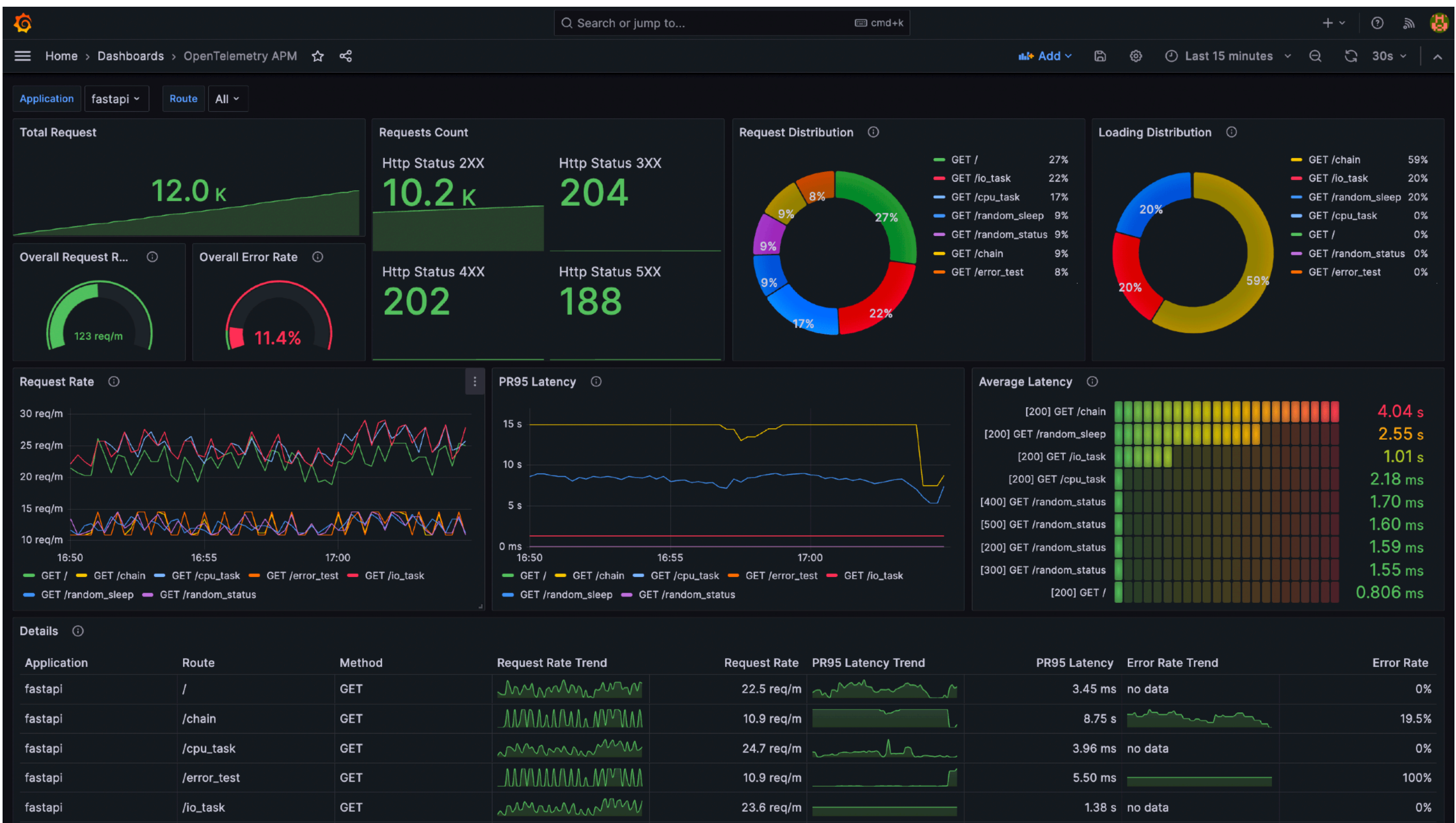
是否散落各處，形成 Data Silo？



Observability Platform - Grafana

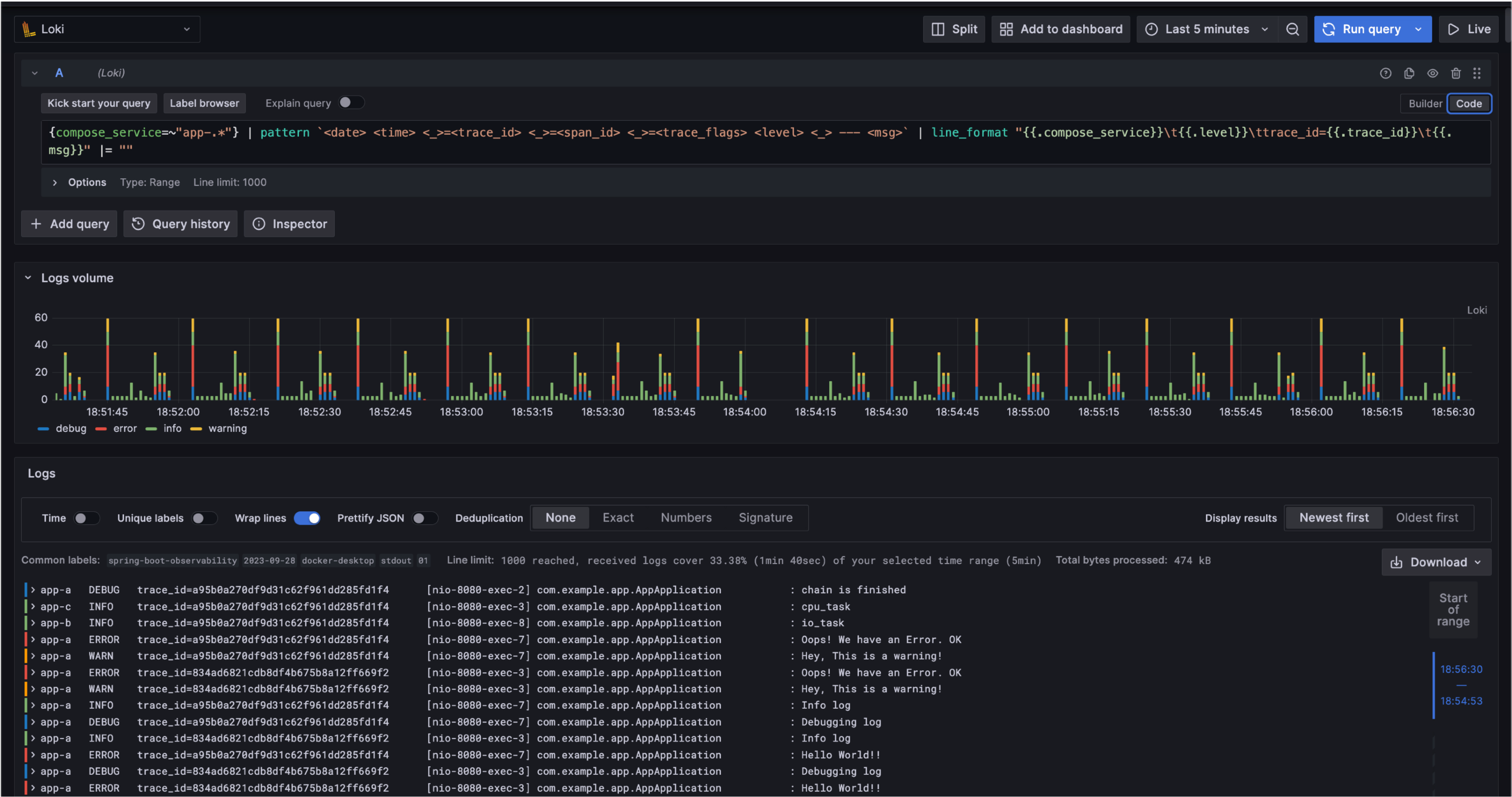


Grafana Labs 開源的資料視覺化與儀表板工具，並提供告警功能



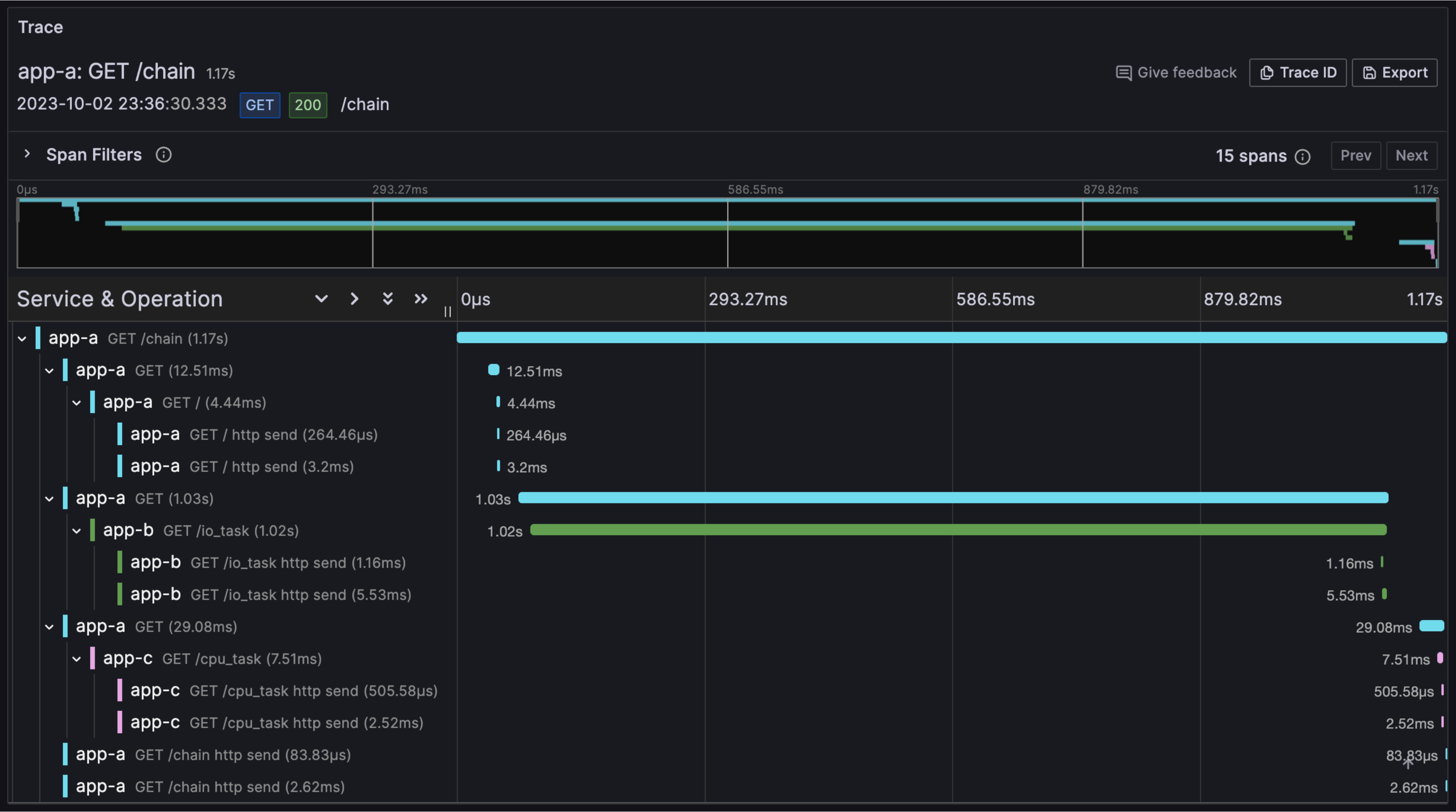
Metrics 儀表板範例 [OpenTelemetry Application Performance Management](#)

Observability Platform - Grafana



Grafana 檢視 Log

Observability Platform - Grafana



Grafana 檢視 Trace

Observability Platform - Grafana

支援各種不同的資料來源 (Data Source)

Prometheus、ElasticSearch、PostgreSQL、Jaeger、Google Sheet



圖片來源：Grafana

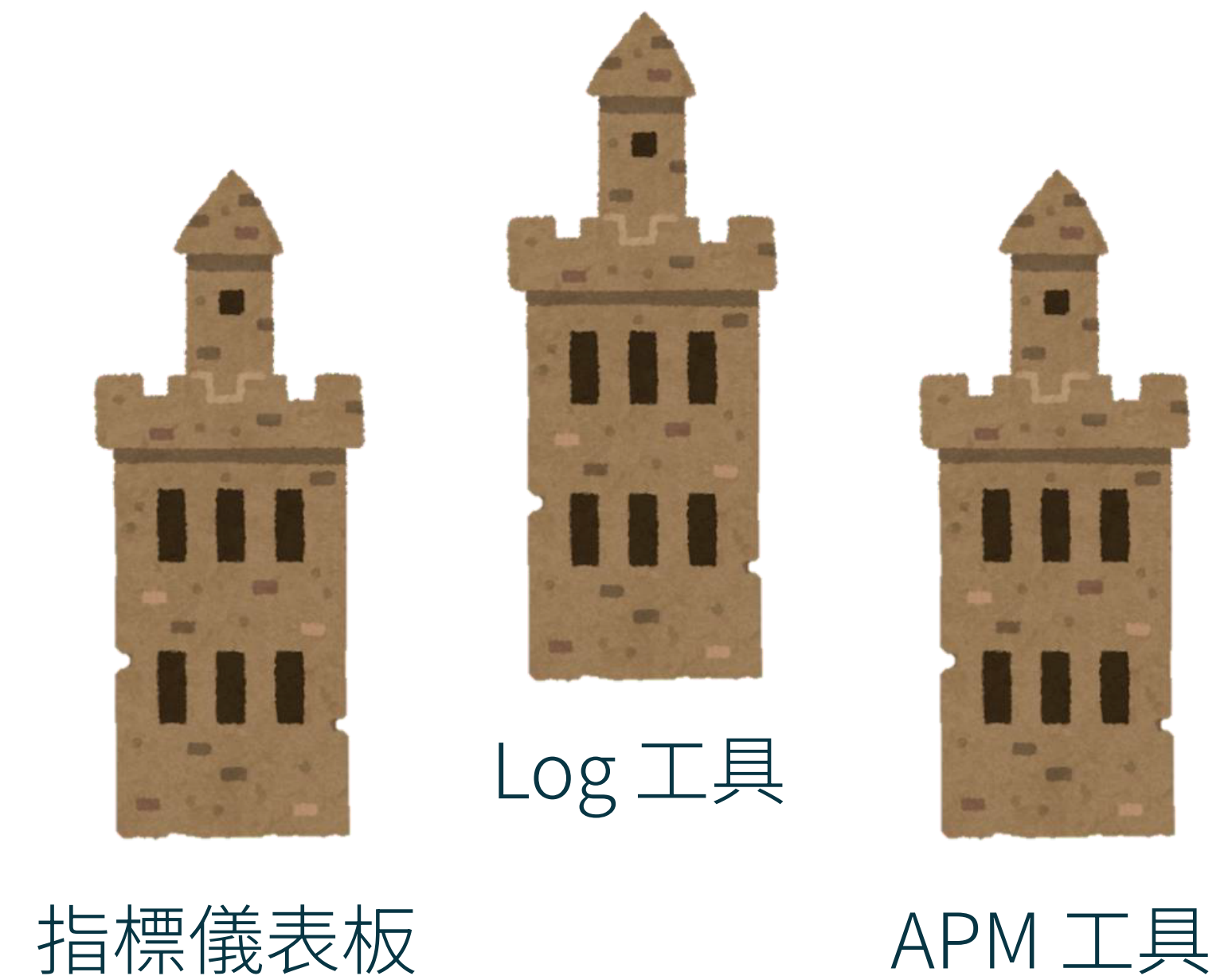
Observability Intro

透過各種資訊，清楚了解系統狀態

有足夠的資訊嗎？

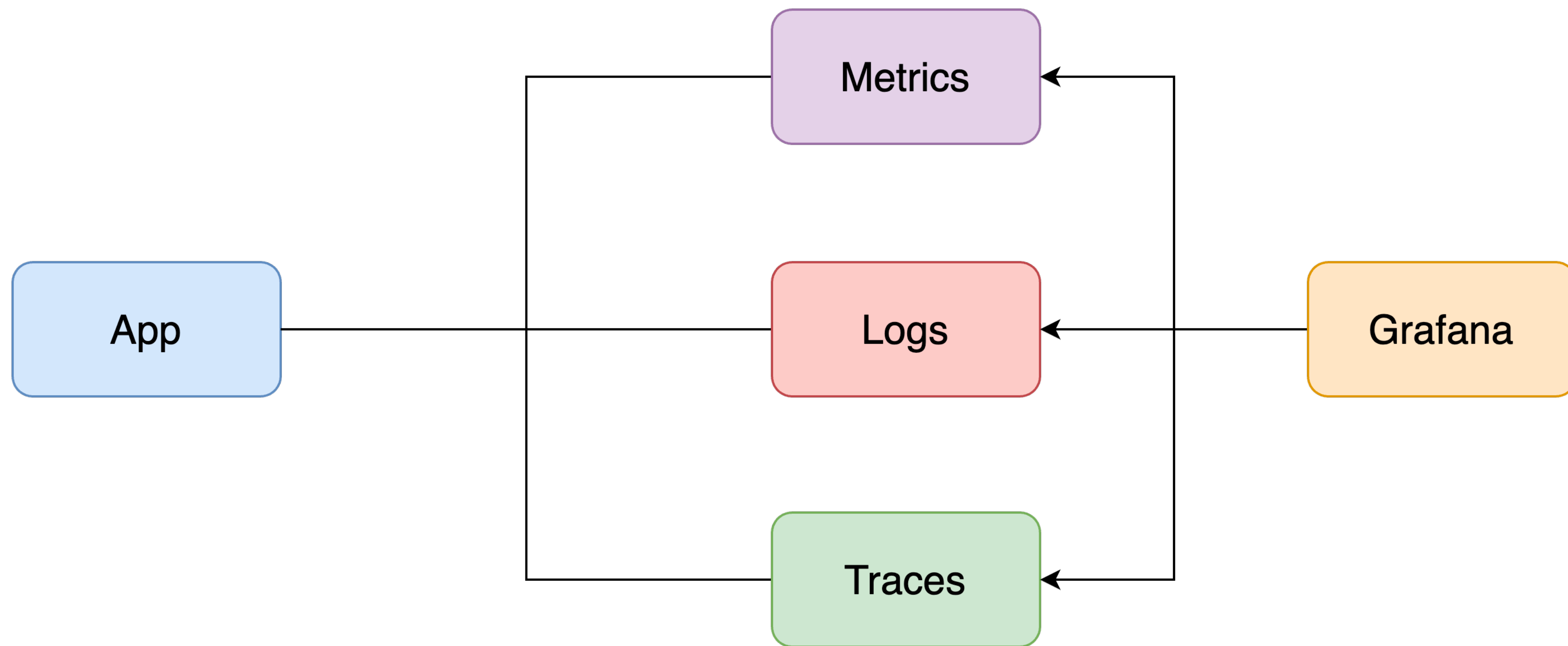


是否散落各處，形成 Data Silo？



Observability Signals - Data Flow

Observability Signals - Data Flow



Observability Signals - Data Flow

資訊處理四步驟

生成



收集



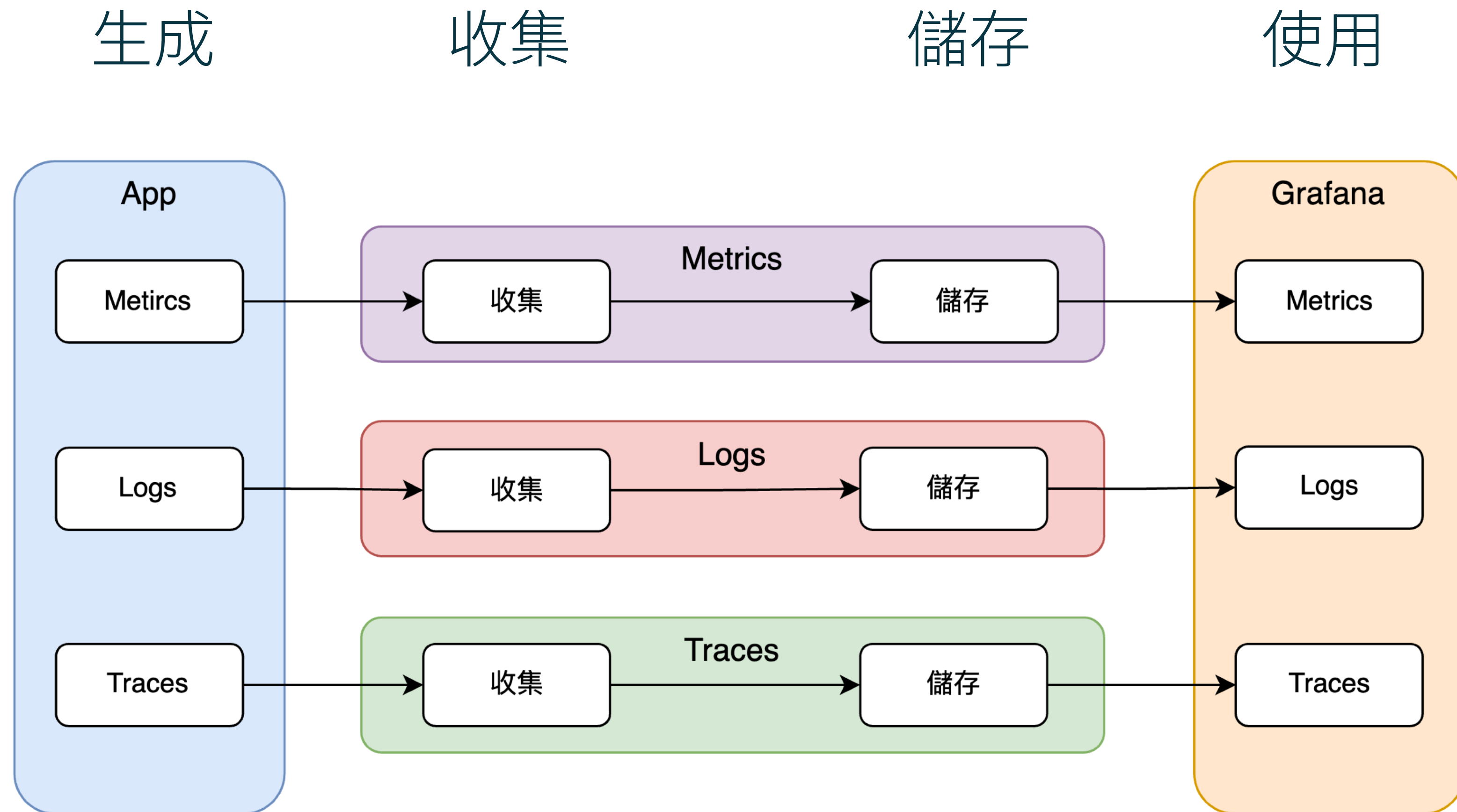
儲存



使用



Observability Signals - Data Flow



了解工具在 Data Flow 中的定位

Observability Signals - Metrics

Metrics - Prometheus



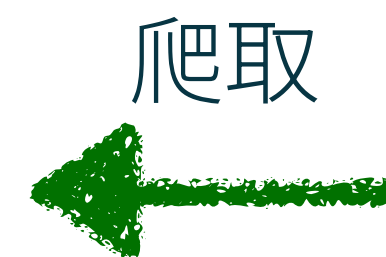
- Prometheus
 - 監控 (Monitoring) 與告警 (Alerting) 工具
 - 2012 年由 SoundCloud 開發並開源
 - 2024 年 11 月進入 3.0 版本
- **收集**不同時間點的量化指標，**儲存**後能以 PromQL 語法**查詢**

生成

收集 儲存 使用

```
# HELP fastapi_requests Total count of requests by method and path.  
# TYPE fastapi_requests counter  
fastapi_requests_total{method="GET",path="/"} 6.0  
fastapi_requests_total{method="GET",path="/metrics"} 647.0  
fastapi_requests_total{method="GET",path="/io_task"} 3.0
```

Prometheus Metrics



爬取



使用 PromQL 查詢



Metrics - 生成：Prometheus Metrics

- Metrics 生成
 - 格式：[Prometheus Metrics](#)
- 系統指標
 - **機器、服務相關**的資訊，如：CPU、Memory、磁碟空間、JVM 資訊
 - 產生指標的工具 ([Exporter](#))
 - [Node Exporter](#)、[cAdvisor](#)、[JMX Exporter](#)、[PostgreSQL Exporter](#)
- 業務指標
 - **應用、業務相關**的資訊，如：Request 頻率、API 回應時間、Error Rate
 - 產生指標的工具
 - Java Spring Boot：[Spring Boot Actuator](#) 搭配 [Micrometer](#)
 - 其他語言：[Prometheus Client Library](#)

Metrics - 生成：Prometheus Metrics

container_memory_usage_bytes{name="garafna"} [value] [unix timestamp]

Metrics Name

Label Pair

```
# HELP container_memory_usage_bytes Current memory usage in bytes, including all memory regardless of when it was accessed
# TYPE container_memory_usage_bytes gauge
container_memory_usage_bytes{image="prom/prometheus:v2.47.2",name="prometheus"} 1.05234432e+08 1699177617022
container_memory_usage_bytes{image="grafana/grafana:10.2.0",name="grafana"} 8.8047616e+07 1699177616426
container_memory_usage_bytes{image="gcr.io/cadvisor/cadvisor:v0.47.2",name="cadvisor"} 4.425728e+07 1699177616430

# HELP logback_events_total Number of log events that were enabled by the effective log level
# TYPE logback_events_total counter
logback_events_total{application="app-c",level="trace",} 0.0
logback_events_total{application="app-c",level="debug",} 1.0
logback_events_total{application="app-c",level="error",} 3.0
logback_events_total{application="app-c",level="warn",} 2.0
logback_events_total{application="app-c",level="info",} 15.0

# HELP fastapi_requests Total count of requests by method and path.
# TYPE fastapi_requests counter
fastapi_requests_total{method="GET",path="/"} 6.0
fastapi_requests_total{method="GET",path="/metrics"} 647.0
fastapi_requests_total{method="GET",path="/io_task"} 3.0
```


Metrics - 收集與儲存：Prometheus

<http://app-a:8000/metrics>



Prometheus

```
# HELP request_processing_seconds Time spent processing request
# TYPE request_processing_seconds histogram
request_processing_seconds_bucket{le="0.005"} 1.0
request_processing_seconds_bucket{le="0.05"} 1.0
request_processing_seconds_bucket{le="0.5"} 1.0
request_processing_seconds_bucket{le="2.5"} 6.0
request_processing_seconds_bucket{le="7.5"} 17.0
request_processing_seconds_bucket{le="+Inf"} 17.0
request_processing_seconds_count 17.0
request_processing_seconds_sum 52.04826683411375
# HELP request_in_processing Requests in progress
# TYPE request_in_processing gauge
request_in_processing 0.0
# HELP requests_count_total Total count of requests
# TYPE requests_count_total counter
requests_count_total 17.0
```

```
prometheus.yml

scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]

  - job_name: 'cadvisor'
    static_configs:
      - targets: ['cadvisor:8080']

  - job_name: 'app-a'
    scrape_interval: 5s
    metrics_path: '/metrics'
    static_configs:
      - targets: ['app-a:8000']
```

Scrape

Metrics - 收集與儲存：Prometheus

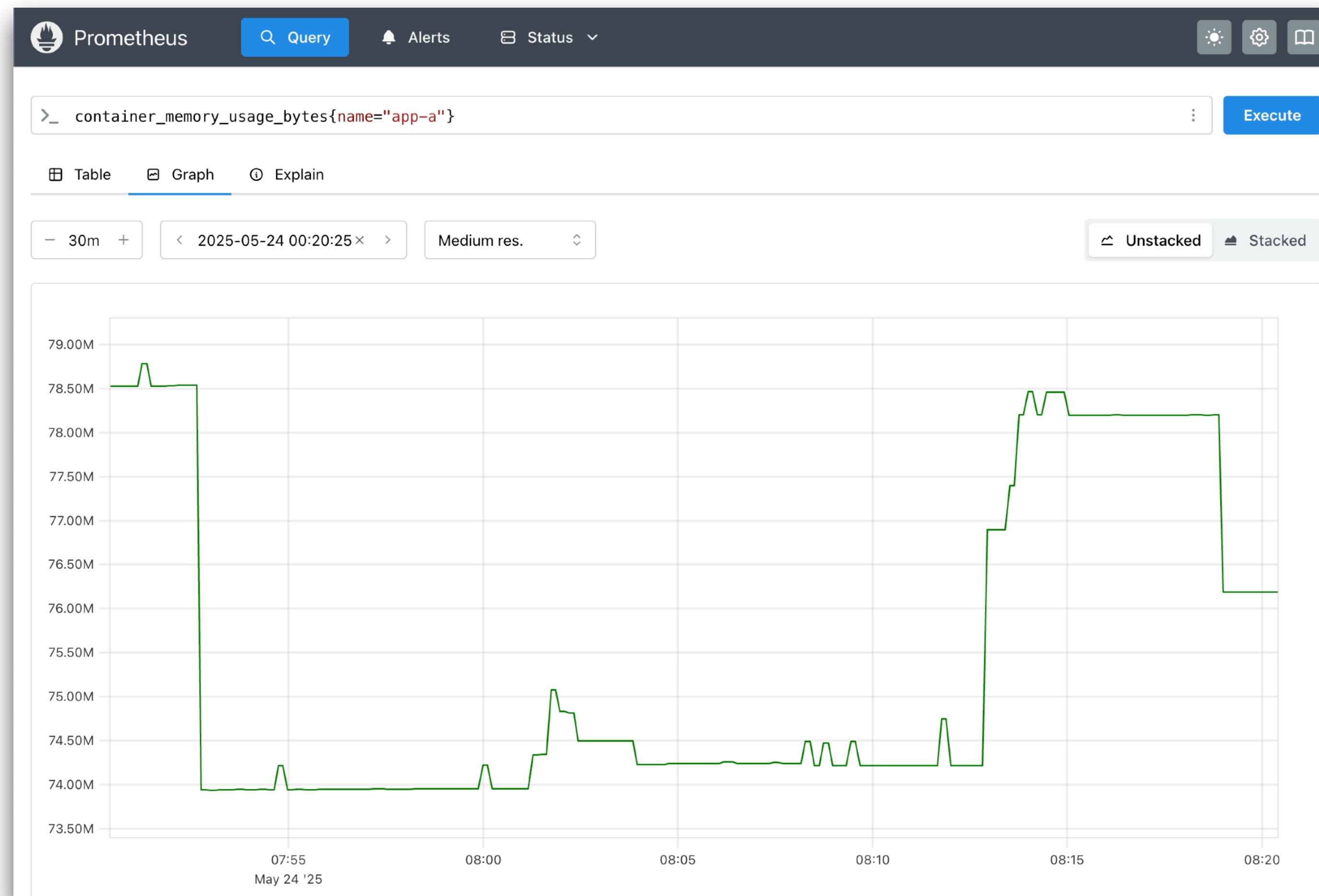
The left screenshot shows the Prometheus Web UI with the 'Status' menu open. The 'Target health' option is highlighted with a red box. A red arrow points from this option to the right screenshot.

The right screenshot shows the 'Target health' page. It displays a table of monitoring targets with columns for Endpoint, Labels, Last scrape, and State.

Endpoint	Labels	Last scrape	State
app-a 1/1 up			
http://app-a:8000/metrics	instance="app-a:8000" job="app-a"	2.851s ago 2ms	UP
app-b 1/1 up			
http://app-b:8000/metrics	instance="app-b:8000" job="app-b"	2.563s ago 6ms	UP
app-c 1/1 up			
http://app-c:8000/metrics	instance="app-c:8000" job="app-c"	4.821s ago 2ms	UP

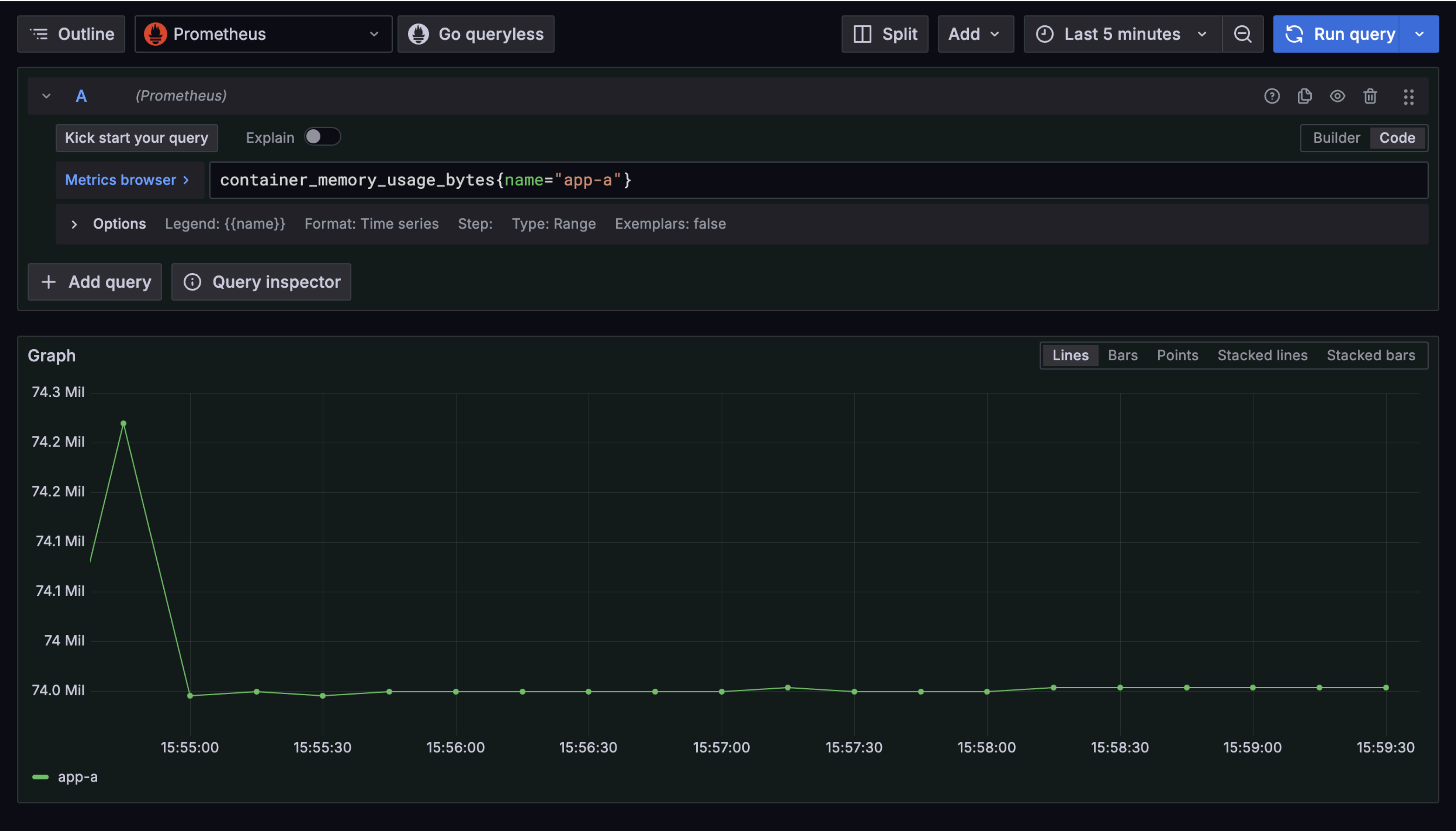
Prometheus Web UI 可以查看設定的爬取目標與爬取狀況

Metrics - 使用：Prometheus Web UI

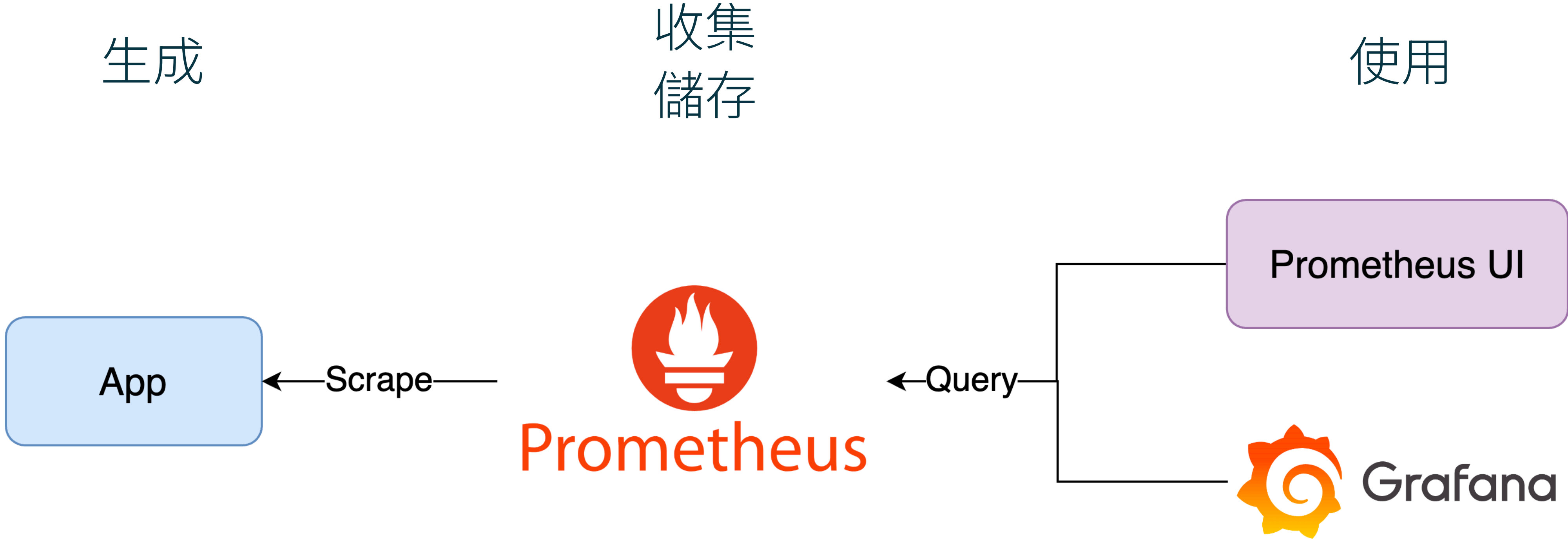


使用 Prometheus Web UI 查詢 Metrics

Metrics - 使用：Prometheus with Grafana



Metrics - Data Flow



Observability Signals - Logs

Logs - 生成

- 核心內容
 - 時間、Level、metadata、訊息
- 常見格式
 - 非結構化，自訂的 Log Pattern
 - 2025-05-30T16:00:00Z INFO [main] Hello World!
 - 結構化，JSON、logfmt
 - {"ts":"2025-05-30T16:00:00Z","level":"INFO","msg":"Hello World!"}
- 輸出方式
 - Console：通過 STDOUT 或 STDERR 輸出至 Console
 - File：儲存在檔案中，可能搭配 Rotate 進行檔案管理
 - 網路傳輸：將 Log 送往遠端儲存服務

Logs - 收集與儲存：Loki & Docker Driver



- Loki
 - Grafana Labs 開源的 Log **儲存** 工具
 - 借鑑 Prometheus 與 PromQL 設計理念，透過 LogQL 查詢 Log
- Loki Docker Driver
 - 使用 Docker 的 Logging Driver 機制，**收集** Container Log 送至 Loki
- 支援 Loki 的收集工具：Fluent Bit、Vector ...

Logs - 收集與儲存：Loki & Loki Docker Driver

```
docker-compose.yml

x-logging: &default-logging
driver: loki
options:
  loki-url: 'http://localhost:3100/api/prom/push'
  loki-pipeline-stages: |
    - multiline:
        firstline: '^\\d{4}-\\d{2}-\\d{2} \\d{1,2}:\\d{2}:\\d{2}'
        max_wait_time: 3s
    - regex:
        expression: '^(?P<time>\\d{4}-\\d{2}-\\d{2} \\d{1,2}:\\d{2}:\\d{2},\\d{3}) (?P<message>(?!s:.*) )$'
```

services:

```
  app:
    image: ghcr.io/blueswen/fastapi-observability/app:latest
    logging: *default-logging
```

Push



Loki Docker Driver 收集 Log 後推送至 Loki

Logs - 使用：Loki with Grafana

OutlineLokiGo querylessSplitAddLast 1 hourRun queryLive

A(Loki)

Kick start your queryLabel browserExplain queryBuilderCode

Label filters

container_name = app-a

Line containsPlease fix this!

+ Operationshint: add logfmt parserhint: add label level format

`{container_name="app-a"} |= 'Please fix this!'`

> OptionsType: RangeLine limit: 1000Direction: Backward

> Logs volume

LogsTable

33 lines displayedTotal bytes processed: 105 kBCommon labels: compose_project=devopsdays-taip... compose_service=app-a container_name=app-a +3

> 2025-06-04 13:38:17,578 CRITICAL [root] [main.py:101] [trace_id=8bc977b0c8f6b9d396837535a2a60719 span_id=630bb2621b960c6a resource.service.name=app-a trace_sampled=Tr
ue] - Critical error. Please fix this!

> 2025-06-04 13:38:16,570 CRITICAL [root] [main.py:101] [trace_id=c1653858cc9c5a231d20697c3ff2bd24 span_id=197d5eb5424ad47b resource.service.name=app-a trace_sampled=Tr
ue] - Critical error. Please fix this!

> 2025-06-04 13:38:15,566 CRITICAL [root] [main.py:101] [trace_id=cd8ef38dde41c360514eefa4bd41c69c span_id=4f6ff8c690035687 resource.service.name=app-a trace_sampled=Tr
ue] - Critical error. Please fix this!

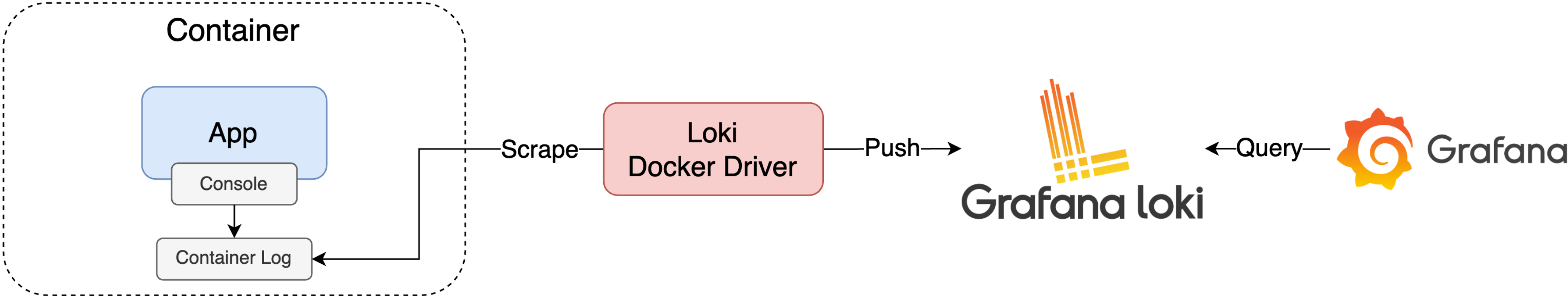
Logs - Data Flow

生成

收集

儲存

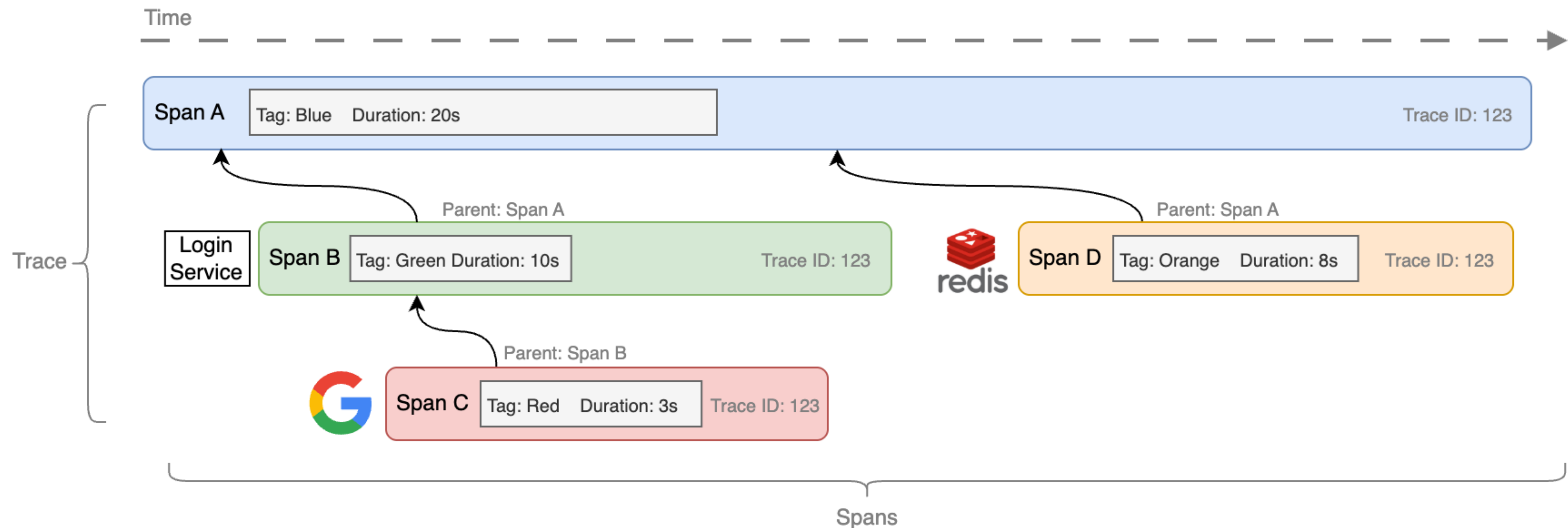
使用



Observability Signals - Traces

Traces(Distributed Tracing)

- 監控**跨服務請求**，利用統一的 **Trace ID** 關聯一個行為在不同服務之間的**歷程**與**資訊**，例如：執行時間、請求來源 IP、SQL 語法、錯誤訊息等



Traces - OpenTelemetry



- OpenTelemetry
 - CNCF Project，目前**主流的 Trace 規範**，已經有超過 80 個組織支援
 - 2019 年由 OpenTracing 與 OpenCensus 合併而成
 - 處理 Telemetry Data (Metrics、Logs、Traces) 的 **API**、**SDK**、**Tool** 組合包

API

資料格式(OTLP)
傳輸方式(HTTP、gRPC)

SDK

開發工具
Instrumentation

Tool

Telemetry Data Collector

Traces - OpenTelemetry



- OpenTelemetry
 - CNCF Project，目前**主流的 Trace 規範**，已經有超過 80 個組織支援
 - 2019 年由 OpenTracing 與 OpenCensus 合併而成
 - 處理 Telemetry Data (Metrics、Logs、Traces) 的 **API**、**SDK**、**Tool** 組合包

API

資料格式(OTLP)
傳輸方式(HTTP、gRPC)

SDK

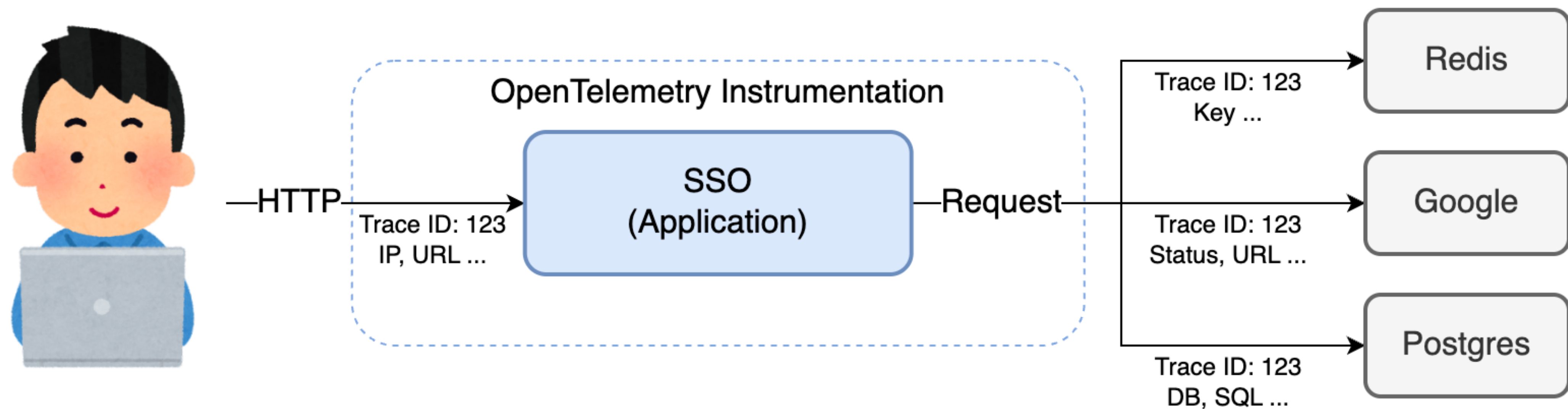
開發工具
Instrumentation

Tool

Telemetry Data Collector

Traces - 生成：OpenTelemetry Instrumentation

- 配備 OpenTelemetry 的 **Instrumentation Package** 後，就像是多了一層收集資料的 **Middleware**，能**自動串連與收集資料**作為 Trace 資訊



Traces - 生成：OpenTelemetry Instrumentation

The screenshot displays a trace for a GET request to the root path. The trace is titled 'basic GET / (5.02s)'. The main span is labeled 'GET /' with a status of 'unset'. The service is 'basic' and the duration is 5.02s. The start time is 0µs (22:07:51.411). The child count is 2 and the kind is 'server'. The library name is 'opentelemetry.instrumentation.fastapi' and the library version is '0.48b0'. The span attributes include http.flavor, http.host, http.method, http.route, http.scheme, http.server_name, http.status_code, http.target, http.url, http.user_agent, net.host.port, net.peer.ip, and net.peer.port. The resource attributes include service.name, telemetry.auto.version, and telemetry.sdk.language.

GET /

Service: basic | Duration: 5.02s | Start Time: 0µs (22:07:51.411) | Child Count: 2 | Kind: server

Status: unset | Library Name: opentelemetry.instrumentation.fastapi | Library Version: 0.48b0

Span Attributes

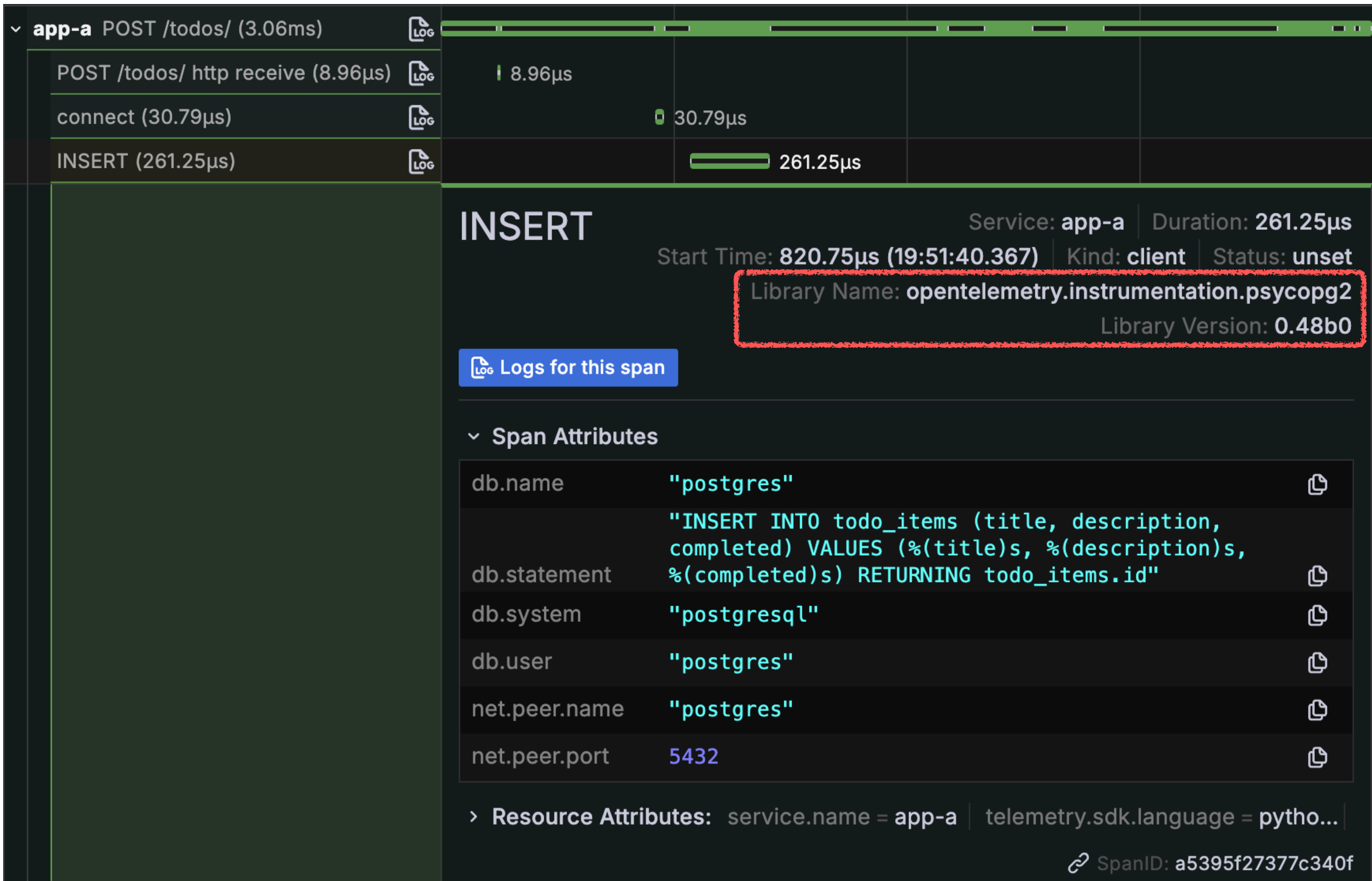
http.flavor	"1.1"
http.host	"172.18.0.7:8080"
http.method	"GET"
http.route	"/"
http.scheme	"http"
http.server_name	"localhost:8080"
http.status_code	200
http.target	"/"
http.url	"http://172.18.0.7:8080/"
http.user_agent	"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
net.host.port	8080
net.peer.ip	"192.168.65.1"
net.peer.port	36449

Resource Attributes: service.name = basic | telemetry.auto.version = 0.48b0 | telemetry.sdk.language = py...

SpanID: 181c58ac9eccd796

FastAPI Instrumentation，紀錄 URL、IP、Status Code 等資訊

Traces - 生成：OpenTelemetry Instrumentation



psycopg2 Instrumentation，紀錄 DB、SQL 相關資訊

Traces - 生成：Instrumentation 使用方式

- Instrumentation 使用方式
 - Code-based Instrumentation：需調整程式，自行啟用 Intrumentator
 - Zero-code Instrumentation：無須調整程式碼，搭配語言、框架的機制，自動注入到程式中，支援多種語言：Python、Java、JavaScript、.NET ...

```
opentelemetry-instrument python main.py
```

```
java -javaagent:path/to/opentelemetry-javaagent.jar -jar myapp.jar
```

Python 與 Java 使用 Zero-code Instrumentation 範例

Traces - 收集與儲存：Tempo

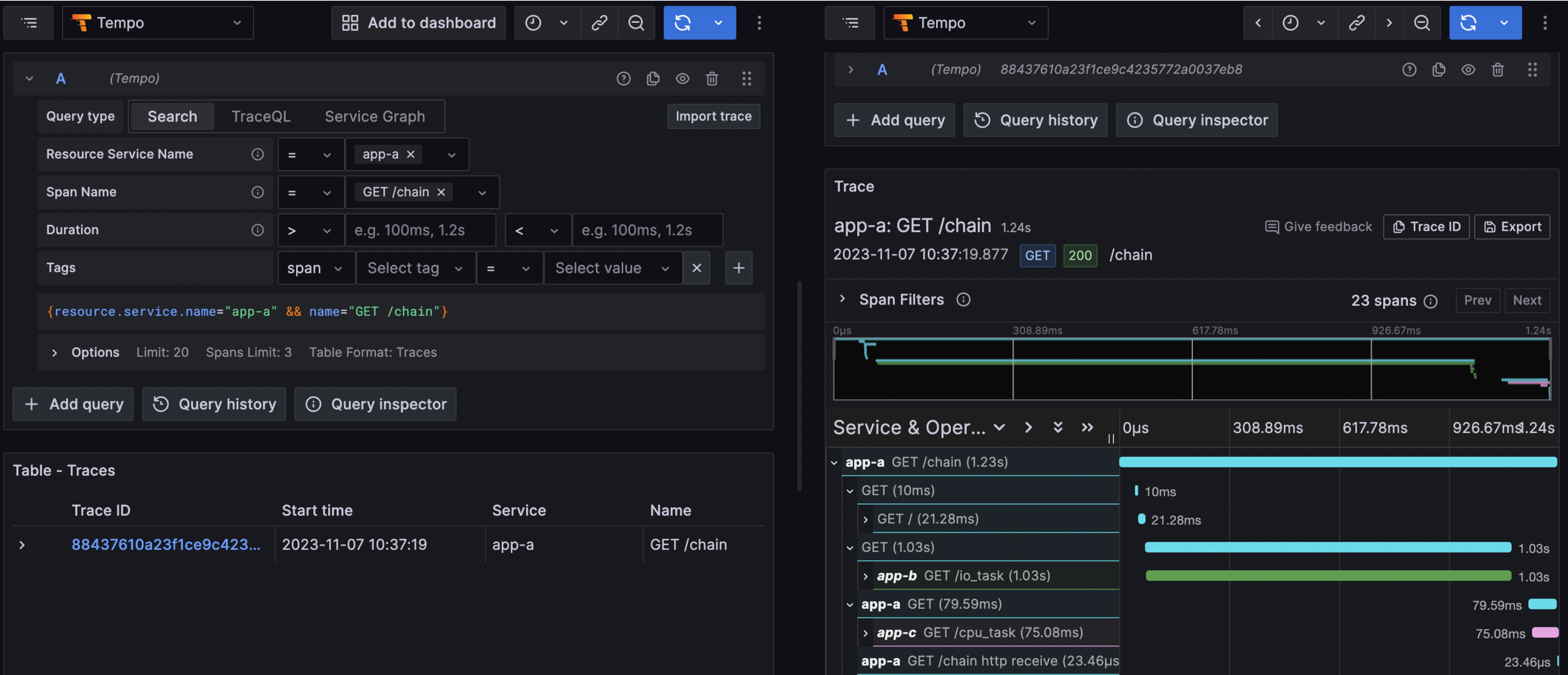


- Tempo
 - Grafana Labs 開源的 Traces 儲存工具
 - 接收 OpenTelemetry Protocol (OTLP) 格式的 Trace 資料

```
app:  
  image: springboot  
  environment:  
    - OTEL_EXPORTER_OTLP_ENDPOINT=http://tempo:4317  
    - OTEL_SERVICE_NAME=app  
  command: "java -javaagent:/opentelemetry-javaagent.jar -jar /app.jar"
```



Traces - 使用：Tempo with Grafana



Traces - Data Flow

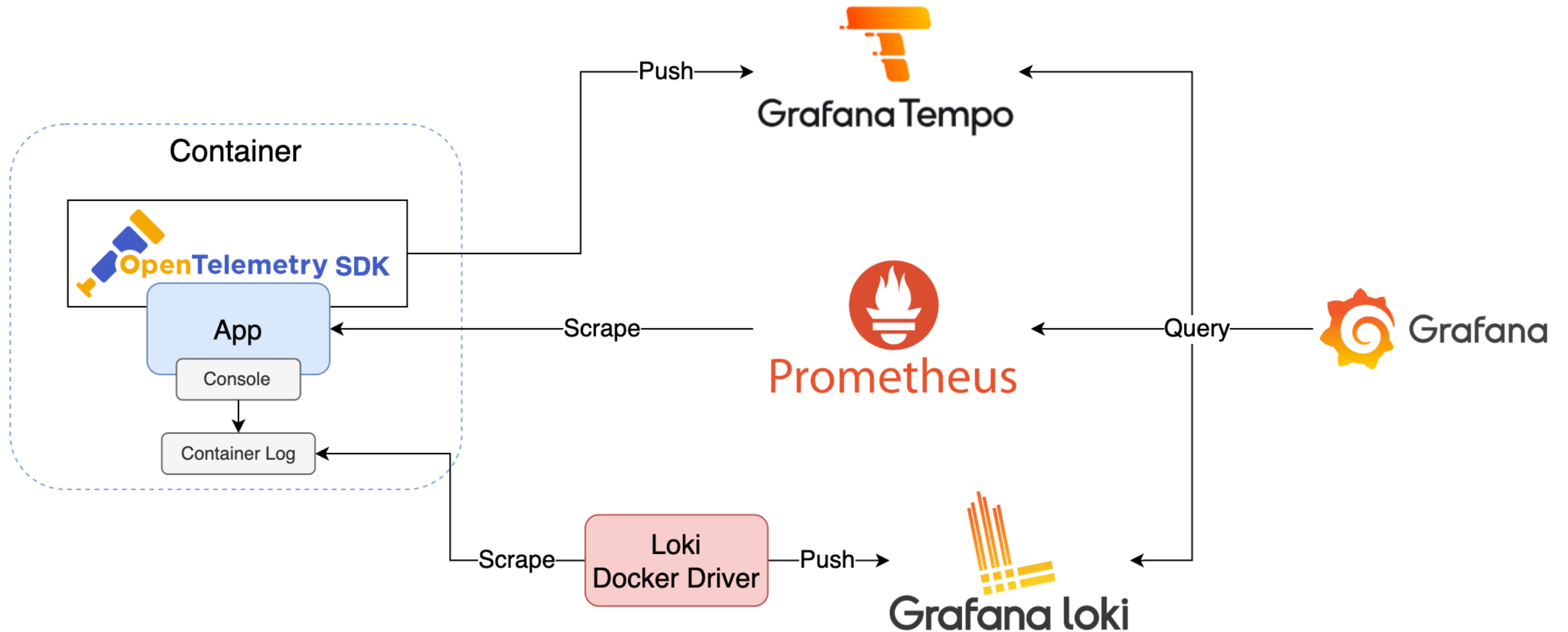
生成

儲存

使用



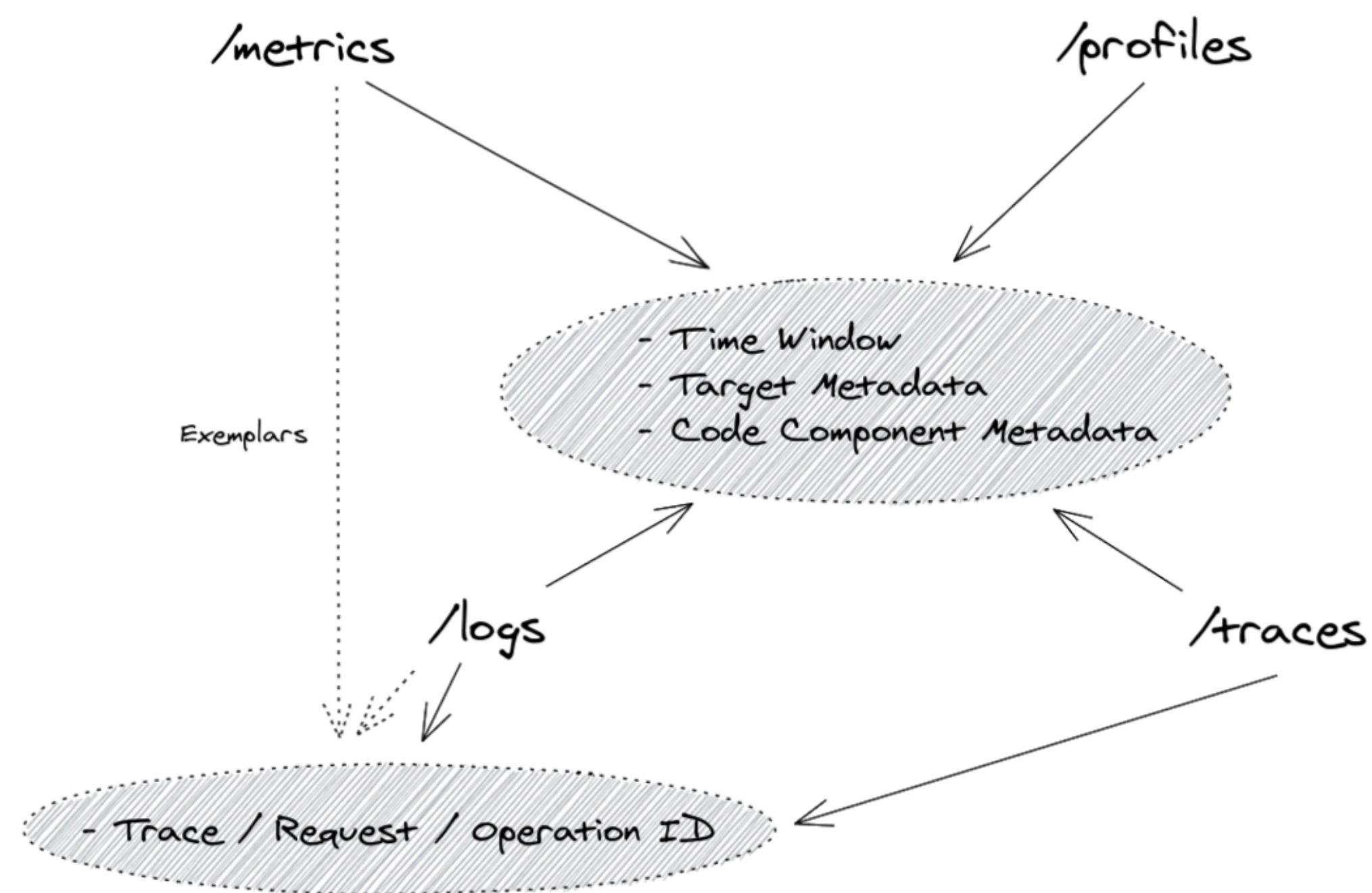
Observability Signals - Data Flow



Observability Signals - 資訊交互應用

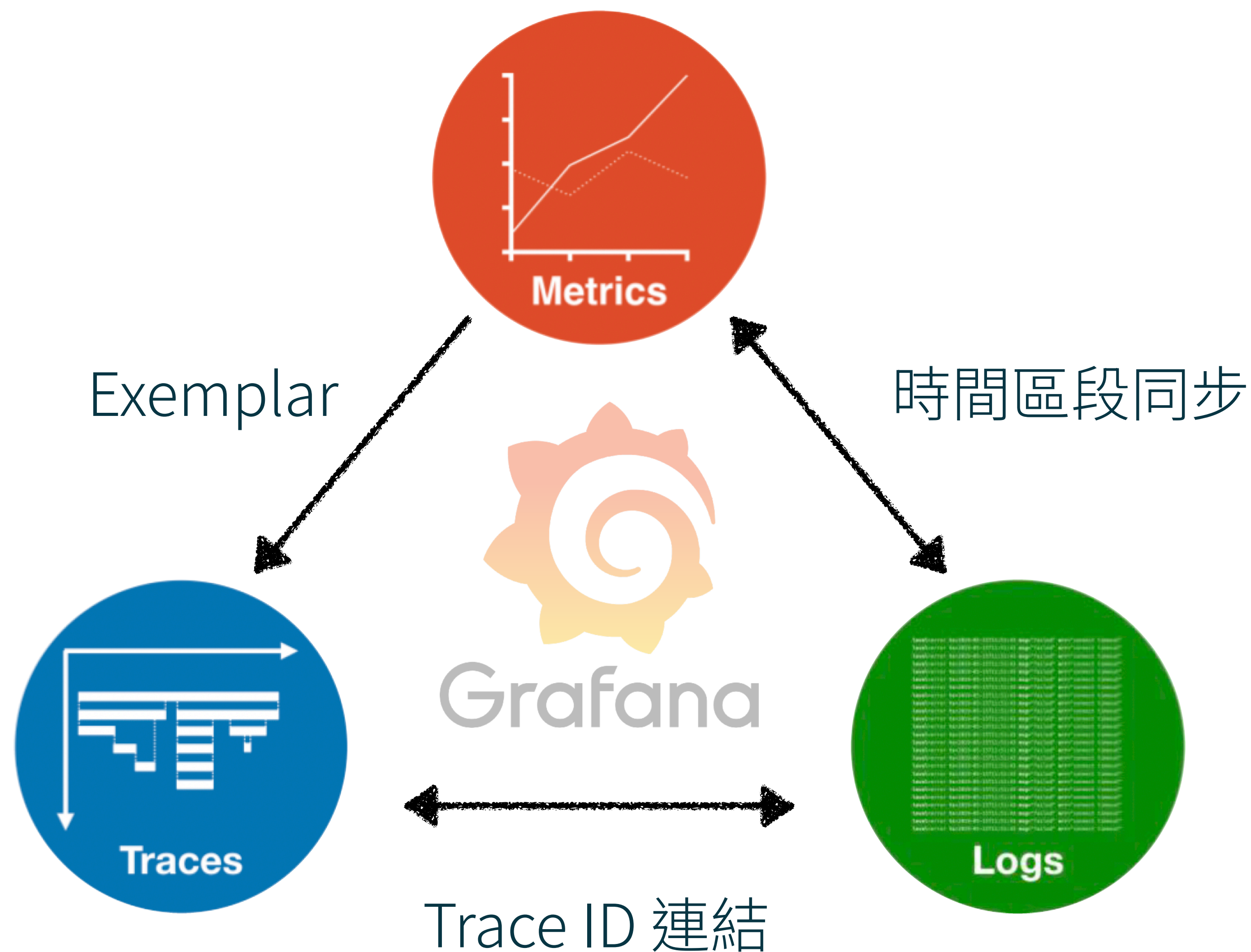
Observability Signals - 資訊交互應用

Observability Signals 彼此具有**關聯**，例如**時間區段**、**Trace ID**
透過**交互應用**產生**綜效**

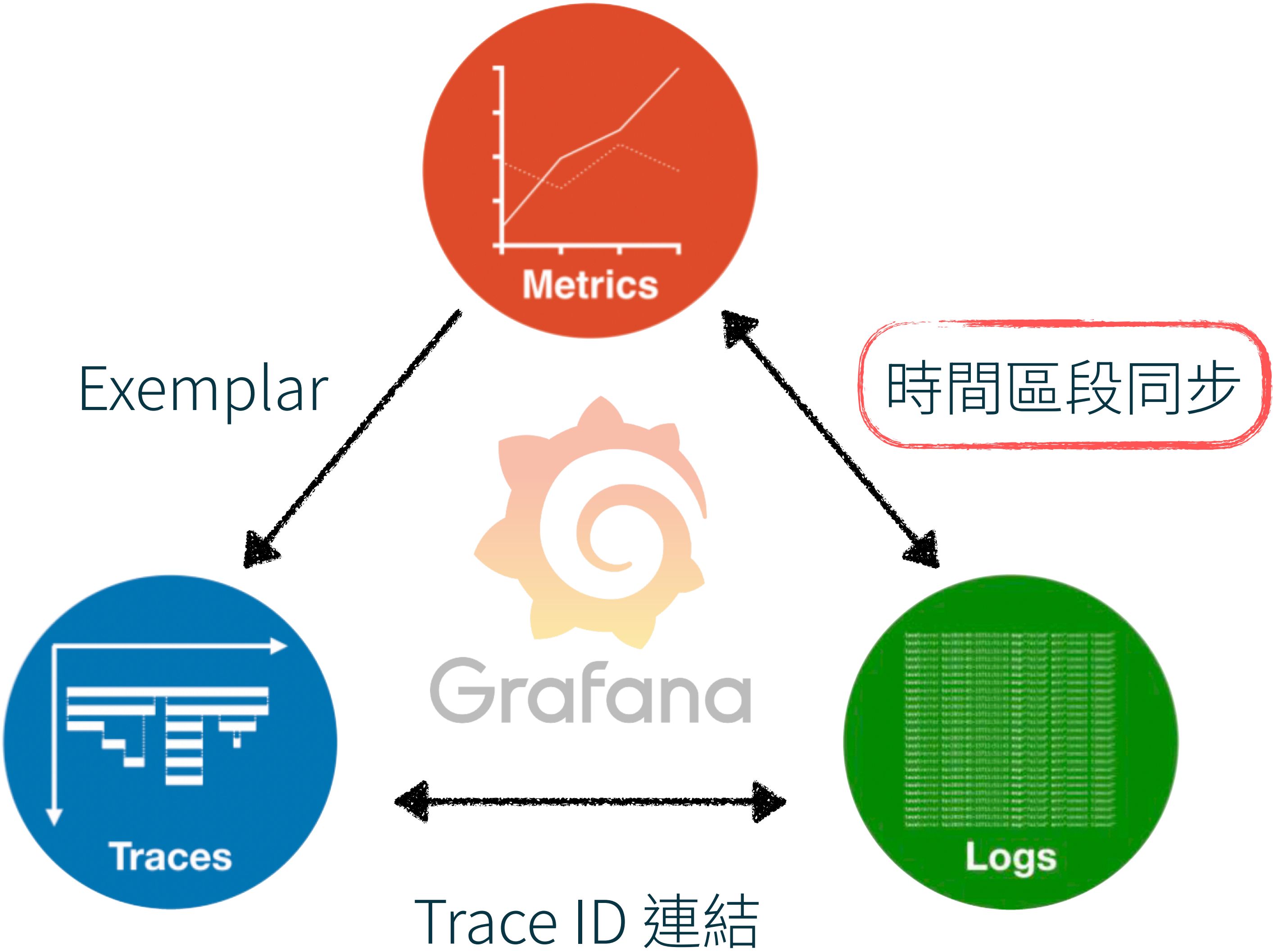


圖片來源：[CNCf Observability Whitepaper](#)

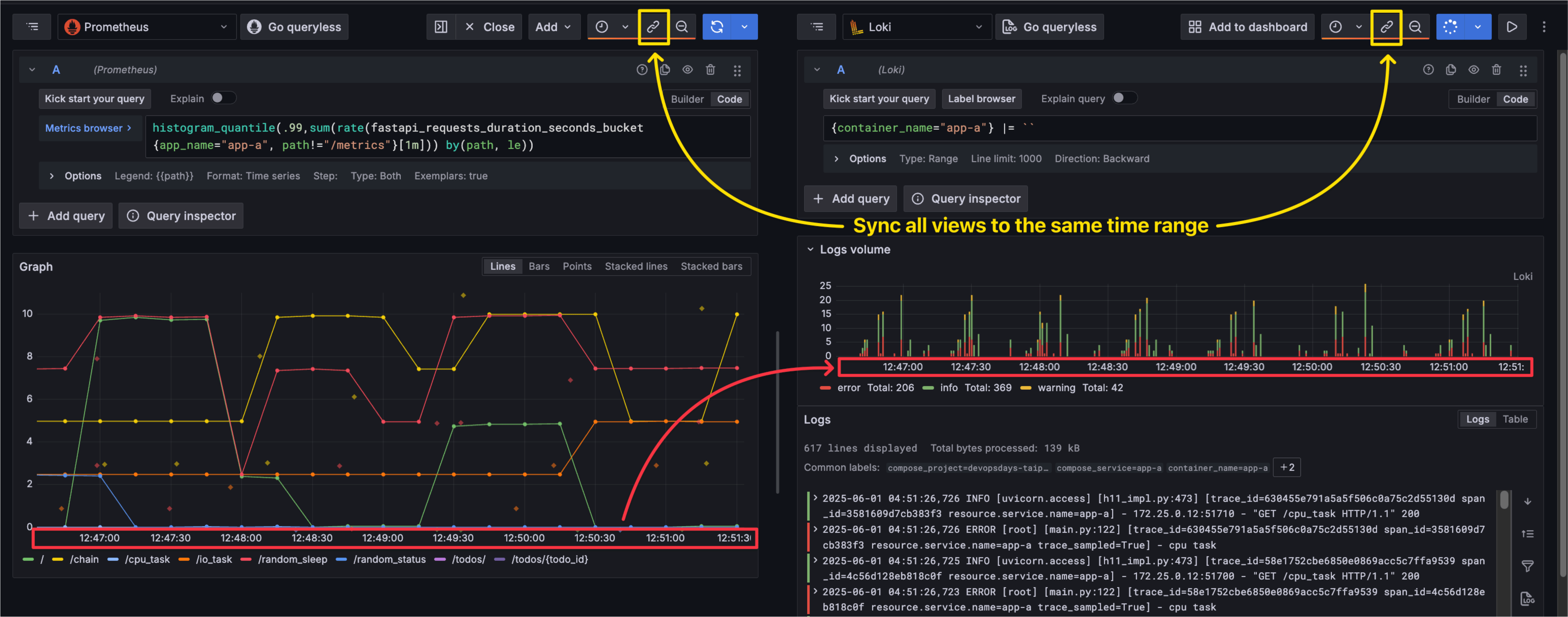
Observability Signals - 資訊交互應用



Observability Signals - 資訊交互應用

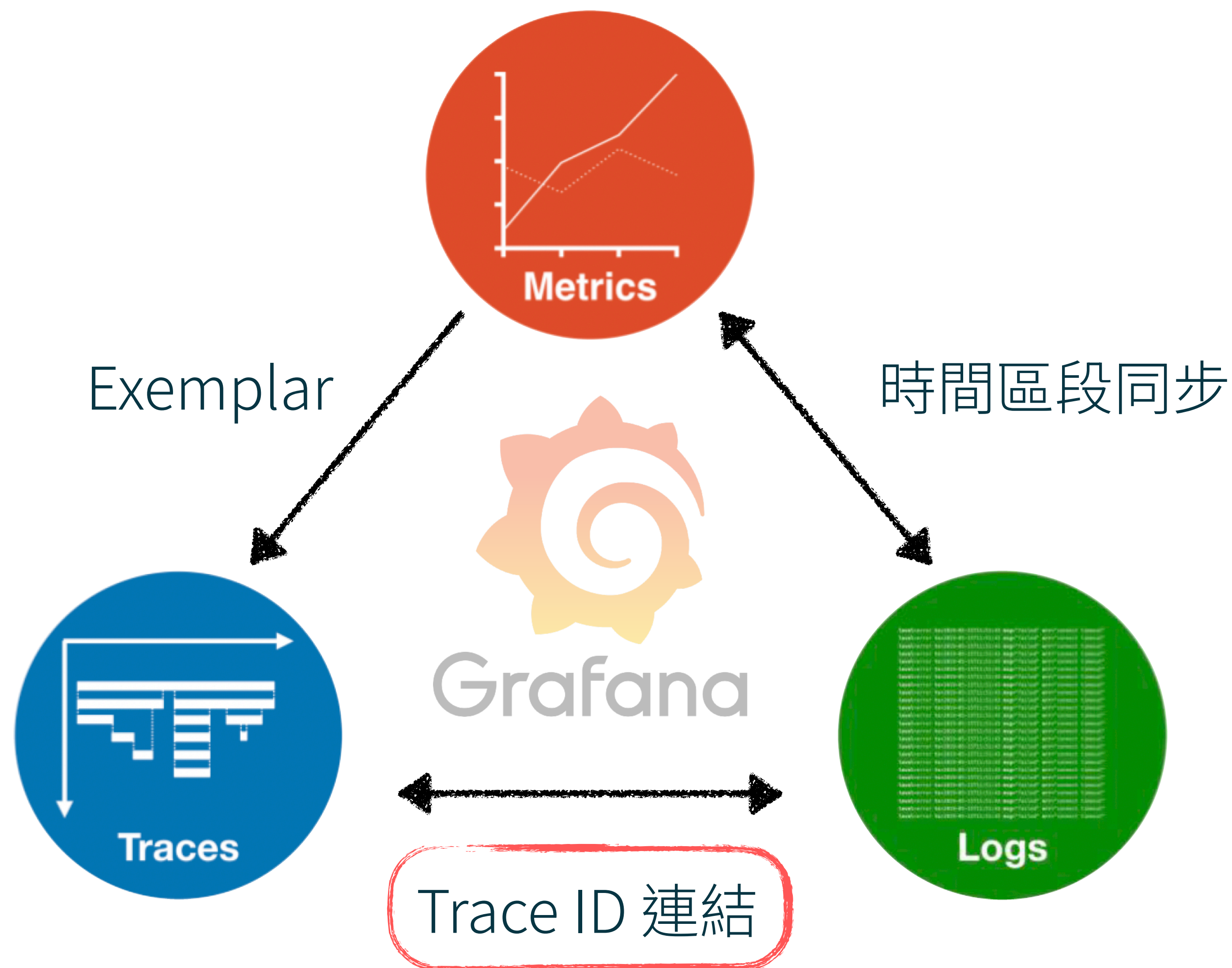


資訊交互應用 - Metrics vs Logs



利用同步時間功能同時查詢 Metrics 與 Logs

Observability Signals - 資訊交互應用



資訊交互應用 - Traces vs Logs

- OpenTelemetry 的 Logging Instrumentation，注入 Trace 資訊到 Log 中

```
> 2025-05-24 08:17:32,396 ERROR [root] [main.py:42] [trace_id=f2dad285651ae77e677e1545dc7e5b66  
span_id=3176a96ef7b5fe6f resource.service.name=app-a trace_sampled=True] - Hello World
```

- Java：搭配 Log4j、Logback 應用 MDC 機制注入

```
logging:  
  pattern:  
    level: "trace_id=%mdc{trace_id} span_id=%mdc{span_id} trace_flags=%mdc{trace_flags} %p"
```

- Python：搭配 Logging Instrumentation

```
from opentelemetry.instrumentation.logging import LoggingInstrumentor  
  
LoggingInstrumentor().instrument(set_logging_format=True)
```

資訊交互應用 - Traces vs Logs

Tempo

Close

Add to dashboard

Search

TraceQL

Service Graph

Import trace

Build complex queries using TraceQL to select a list of traces.

Documentation

2363f3d85d7758a3f62cbb7abbd7c36f

Search Options

Limit: 20

Spans Limit: 3

Table Format: Traces

Streaming: Disabled

Metrics Options

Step: auto

Type: Range

Streaming: Disabled

+ Add query

Query inspector

Node graph

Trace

app-a: GET /chain 8.15s

2025-06-01 12:47:59.924 /chain

Give feedback

Trace ID

Export

Span Filters

19 spans

Prev

Next

0µs

2.04s

4.08s

6.11s

8.15s

Service & Operation

app-a GET /chain (8.15s)

GET /chain

Service: app-a

Duration: 8.15s

Start Time: 0µs (12:47:59.924)

Child Count: 6

Kind: server

Status: unset

Library

entelemetry.instrumentation.fastapi

Library Version: 0.54b1

Log

Logs for this span

Span attributes: http.flavor = 1.1 http.host = 172.25.0.3:8000 http.meth...

Resource attributes: service.name = app-a telemetry.sdk.language = p...

SpanID: e5278bd1f82193c7

Loki

Add to dashboard

Search

Builder

Code

Kick start your query

Label browser

Explain query

{compose_service="app-a" | label_format log_line_contains_trace_id={`{ contains "2363f3d85d7758a3f62cbb7abbd7c36f" \_\_line\_\_ }`} | log_line_contains_trace_id="true" OR trace_id="2363f3d85d7758a3f62cbb7abbd7c36f"}

Options

Type: Range

Line limit: 1000

Direction: Backward

This query will process approximately 7.0 KiB.

+ Add query

Query inspector

Logs volume

Logs

Table

12 lines displayed

Total bytes processed: 552 kB

Common labels: compose_project=devopsdays-taip_ compose_service=app-a container_name=app-a +3

> 2025-06-01 04:48:08,055 INFO [uvicorn.access] [h11_impl.py:473] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=e5278bd1f82193c7 resource.service.name=app-a] - 172.25.0.15:51860 - "GET /chain HTTP/1.1" 200

> 2025-06-01 04:48:08,019 INFO [root] [main.py:164] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=e5278bd1f82193c7 resource.service.name=app-a trace_sampled=True] - Chain End

> 2025-06-01 04:48:08,016 INFO [httpx] [_client.py:1740] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=e5278bd1f82193c7 resource.service.name=app-a trace_sampled=True] - HTTP Request: GET http://app-c:8000/cpu_task "HTT P/1.1 200 OK"

> 2025-06-01 04:48:02,932 INFO [httpx] [_client.py:1740] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=e5278bd1f82193c7 resource.service.name=app-a trace_sampled=True] - HTTP Request: GET http://app-b:8000/io_task "HTT P/1.1 200 OK"

> 2025-06-01 04:47:59,952 INFO [httpx] [_client.py:1740] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=e5278bd1f82193c7 resource.service.name=app-a trace_sampled=True] - HTTP Request: GET http://localhost:8000/ "HTTP/1.1 200 OK"

> 2025-06-01 04:47:59,951 INFO [uvicorn.access] [h11_impl.py:473] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=45b314d9f47b79d0 resource.service.name=app-a] - 127.0.0.1:48730 - "GET / HTTP/1.1" 200

> 2025-06-01 04:47:59,951 CRITICAL [root] [main.py:101] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=45b314d9f47b79d0 resource.service.name=app-a trace_sampled=True] - Critical error. Please fix this!

> 2025-06-01 04:47:59,951 ERROR [root] [main.py:100] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=45b314d9f47b79d0 resource.service.name=app-a trace_sampled=True] - Oops! We have an Error. OK

> 2025-06-01 04:47:59,951 WARNING [root] [main.py:99] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=45b314d9f47b79d0 resource.service.name=app-a trace_sampled=True] - Hey, This is a warning!

> 2025-06-01 04:47:59,951 INFO [root] [main.py:98] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=45b314d9f47b79d0 resource.service.name=app-a trace_sampled=True] - Info log

> 2025-06-01 04:47:59,951 INFO [root] [main.py:96] [trace_id=2363f3d85d7758a3f62cbb7abbd7c36f span_id=45b314d9f47b79d0 resource.service.name=app-a trace_sampled=True] - Request headers: Headers({'host': 'localhost:8000', 'accept': '*/', 'accept-encoding': 'gzip, deflate', 'connection': 'keep-alive', 'user-agent': 'python-httpx/0.28.

點擊 Span 中的 Log 按鈕可查詢對應 Log

資訊交互應用 - Traces vs Logs

The image shows two side-by-side panels in a Grafana dashboard. The left panel is the Loki log viewer, and the right panel is the Tempo trace viewer.

Loki Panel (Left):

- Query: `{compose_service="app-a"}`
- Options: Type: Range, Line limit: 1000, Direction: Backward. Note: This query will process approximately 5.7 KiB.
- Logs volume: 42 lines displayed, Total bytes processed: 386 kB.
- Common labels: `compose_project=devopsdays-taip_`, `compose_service=app-a`, `container_name=app-a`.
- Log entry: `2025-06-01 04:48:08.056 INFO [root] [main.py:151] [trace_id=a7cc6243500ef25a4f3fce26bea2c51e span_id=54af3d6898038b5 resource.service.name=app-a trace_sampled=True] - /chain Start`
- Fields: `compose_project`, `compose_service`, `container_name`, `detected_level`, `filename`, `host`, `service_name`, `source`.
- Links: TraceID `a7cc6243500ef25a4f3fce26bea2c51e` and a `Tempo` button.

Tempo Panel (Right):

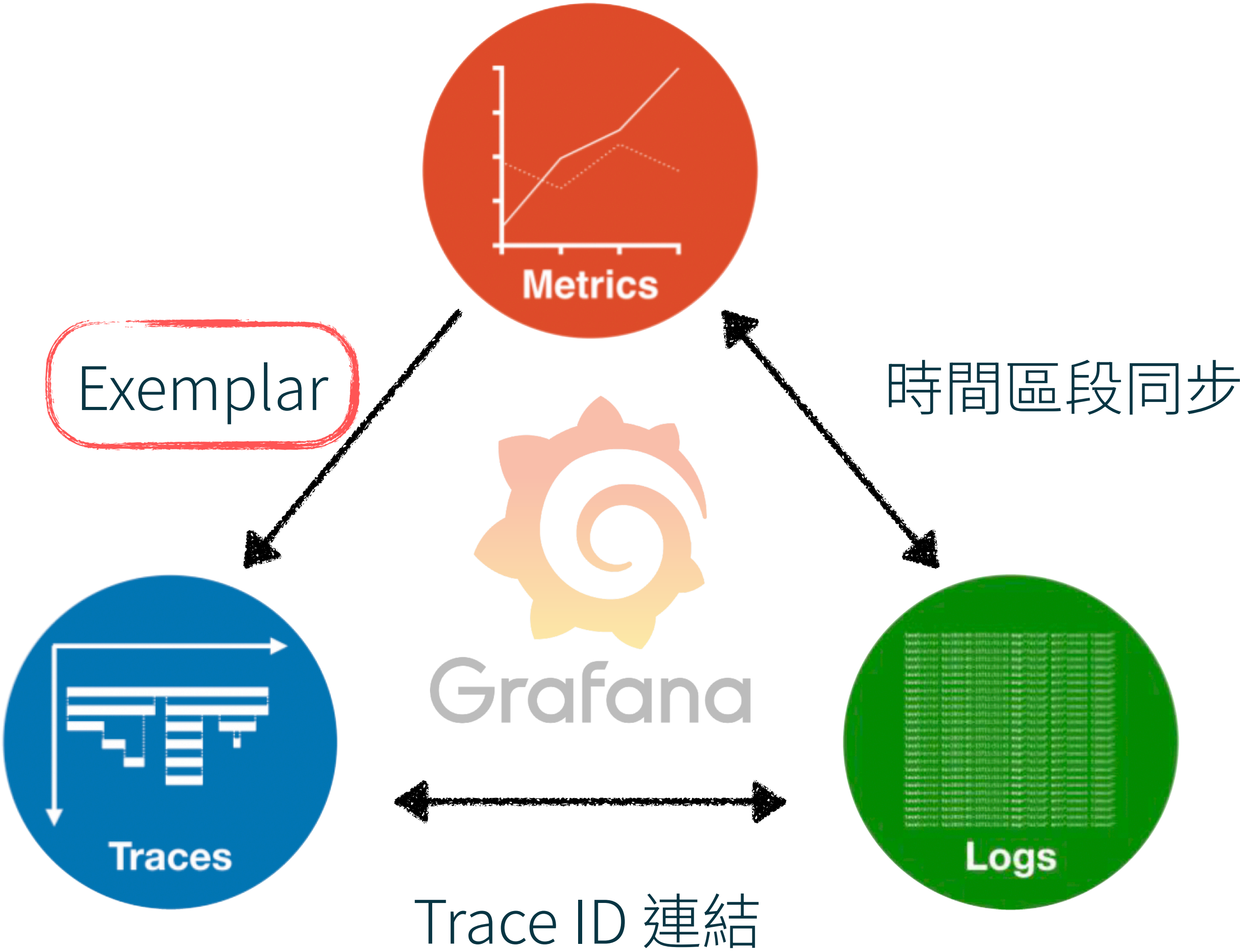
- Query type: Search, TraceQL, Service Graph.
- Build complex queries using TraceQL to select a list of traces.
- Search Options: Limit: 20, Spans Limit: 3, Table Format: Traces, Streaming: Disabled.
- Node graph: app-a: GET /chain 3.12s.
- Span Filters: 19 spans.
- Service & Operation:

Service & Operation	0µs	779.08ms	1.56s	2.34s	3.12s
app-a GET /chain (3.12s)	[Timeline bar]				
GET (25.94ms)	[Timeline bar]				
GET / (2.05ms)	[Timeline bar]				
GET / http send (23.96µs)	[Timeline bar]				
GET / http send (12.33µs)	[Timeline bar]				
GET / http send (7.33µs)	[Timeline bar]				
GET (3s)	[Timeline bar]				
app-b GET /io_task (2.01s)	[Timeline bar]				
GET /io_task http send (31.71µs)	[Timeline bar]				

A red arrow points from the `trace_id` in the log entry to the `a7cc6243500ef25a4f3fce26bea2c51e` trace ID in the Tempo panel. A yellow arrow points from the `Tempo` button in the Loki panel to the same trace ID.

展開 Log 明細點擊按鈕可查看該筆 Trace

Observability Signals - 資訊交互應用



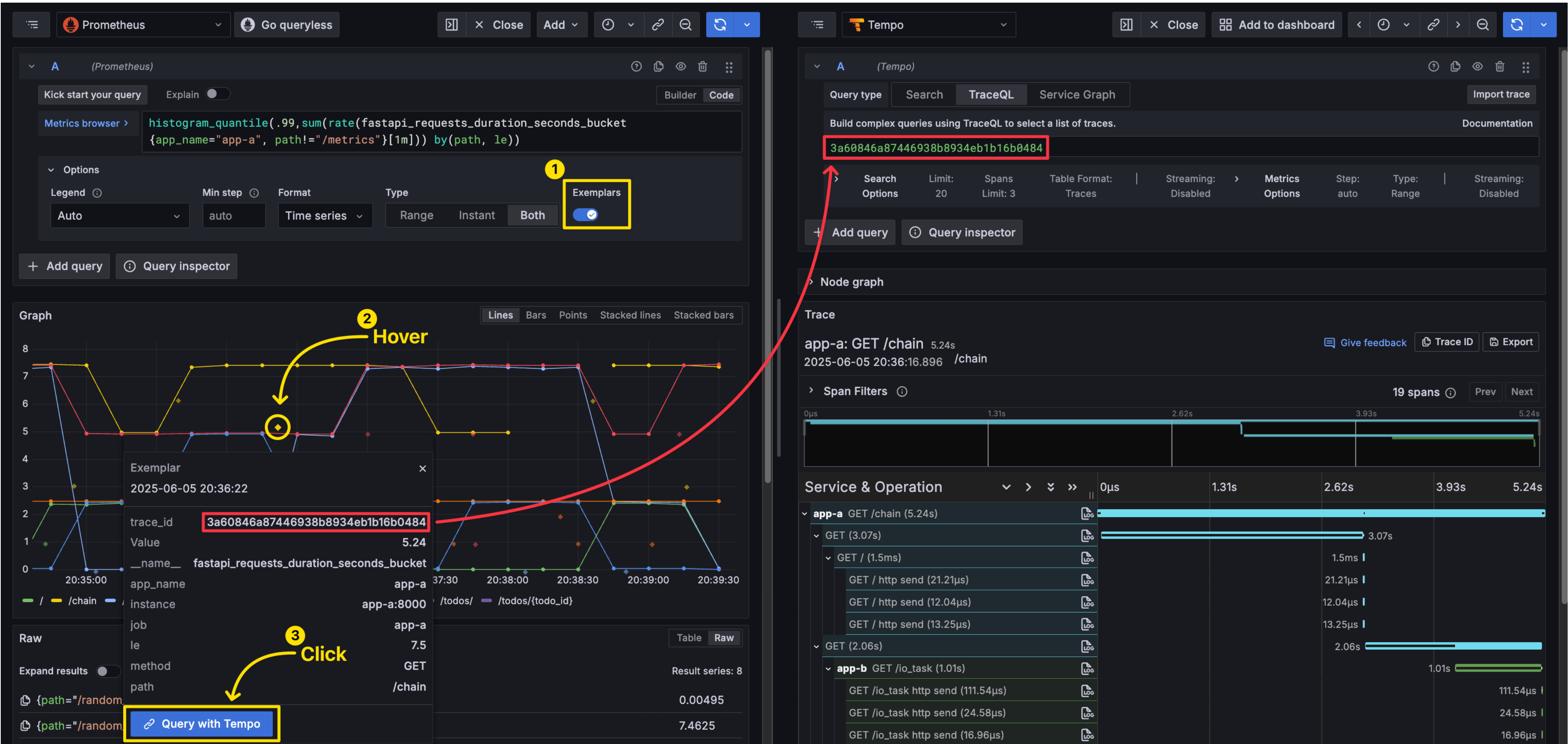
資訊交互應用 - Exemplar: Metrics vs Traces

- Exemplar：在 Metrics 後附上一筆資料，紀錄 Trace ID 與 Metrics 數值
 - 例如：Trace ID 5566 API request duration 為 20 秒
 - Prometheus 需要額外啟用 Exemplar 功能後，才會收集與儲存 Exemplar

```
fastapi_requests_duration_seconds_bucket{app_name="app-a",le="5.0",method="GET",path="/random_sleep"} 1.0 #  
{trace_id="124eb9a05e218ddf9bbe1aeac85bfaca"} 4.013615251998999 1749126722.889076  
fastapi_requests_duration_seconds_bucket{app_name="app-a",le="7.5",method="GET",path="/random_sleep"} 2.0 #  
{trace_id="121a53bfff62845e85ef43a067568c154"} 5.034878544000094 1749126727.9232254  
fastapi_requests_duration_seconds_bucket{app_name="app-a",le="10.0",method="GET",path="/random_sleep"} 3.0 #  
{trace_id="ed4198545fed8e23a0de1997a263b386"} 8.038171295003849 1749126727.9216216
```

帶有 Trace ID 的 Metrics

資訊交互應用 - Exemplar: Metrics vs Traces



透過 Exemplar 的 Trace ID，連結 Metrics 與 Traces

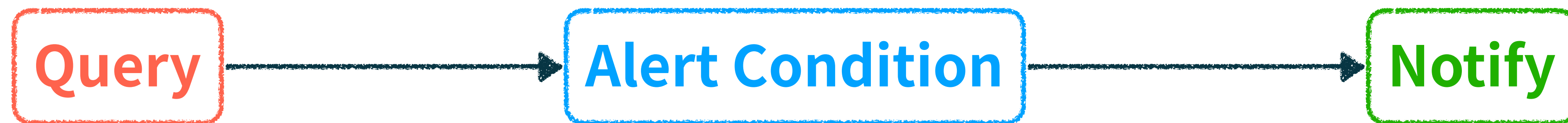
Hands-on - Lab 01: Observability 平台

- GitHub URL : [Lab 01](#)
- Goals : 檢視 Grafana 設定，操作交互應用
- Tasks
 - 開啟 <http://localhost:8000>，在 Explore 中查詢 App A 的 Log
 - 開啟 <http://localhost:8000/chain>，在 Explore 中查詢 App A 該筆 Request 的 Trace
 - 執行 k6-script-todo.js，檢視 FastAPI Observability Dashboard，排查 App A /todos/ 時間偏高的原因

Alerting

Alerting - Grafana Alerting

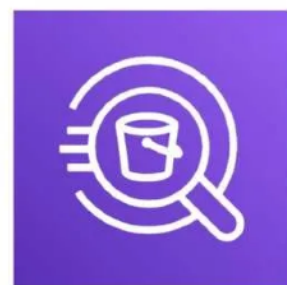
Grafana Alerting 支援多種資料來源與通知方式



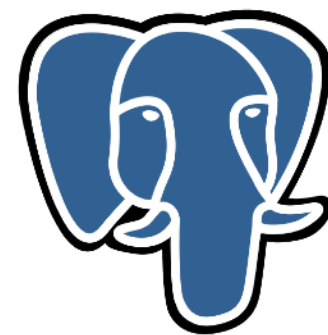
Prometheus



Grafana loki



Amazon Athena



PostgreSQL



elasticsearch

CPU 使用率 > 90%
記憶體用量 > 1 GB
回應時間 > 3s



Telegram

slack



Discord



kafka



webhooks

Alerting - Grafana Alerting

不是所有 Data Source 都支援 Alerting

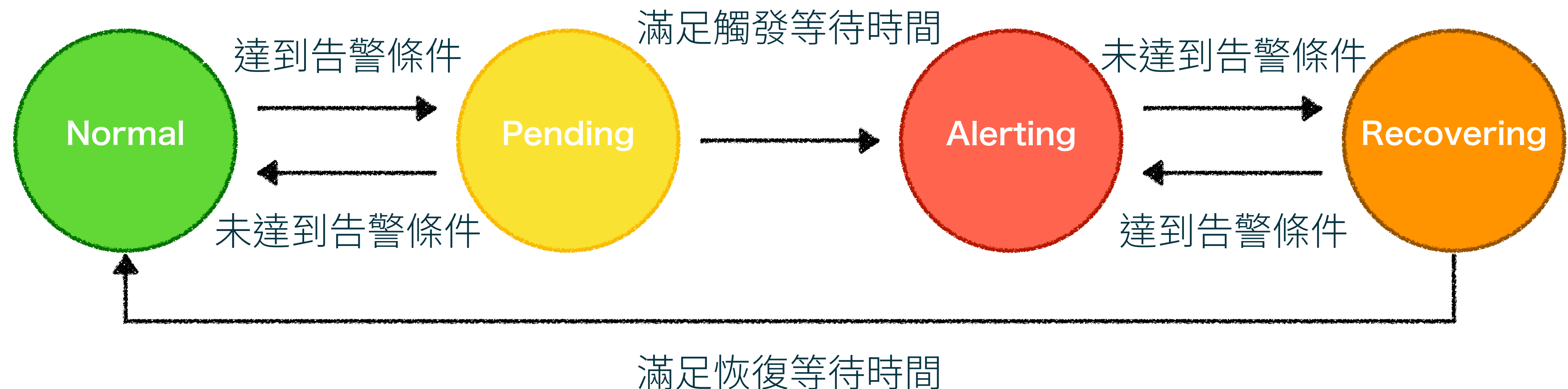
The image displays three screenshots of the Grafana web interface, specifically the 'Data sources' page. Each screenshot shows a different data source and its alerting status. In each case, the 'Alerting' status is highlighted with a red dashed box.

- Loki:** The first screenshot shows the 'Loki' data source. The 'Alerting' status is 'Supported' (green box).
- Prometheus:** The second screenshot shows the 'Prometheus' data source. The 'Alerting' status is 'Supported' (green box).
- Tempo:** The third screenshot shows the 'Tempo' data source. The 'Alerting' status is 'Not supported' (red box).

Each screenshot includes a breadcrumb trail (Home > Connections > Data sources > [Data Source]), a search bar, and buttons for 'Explore data' and 'Build a dashboard'.

Alerting - Alert State

- Grafana Alerting 定期執行查詢，達到告警條件時轉換 Alert State
 - Pending Period：持續超過指定時長才觸發告警
 - Recovering Period*：持續超過指定時長才恢復正常



Alerting - Alert Rule

告警條件：CPU 使用率 > 90%

查詢週期：5 min

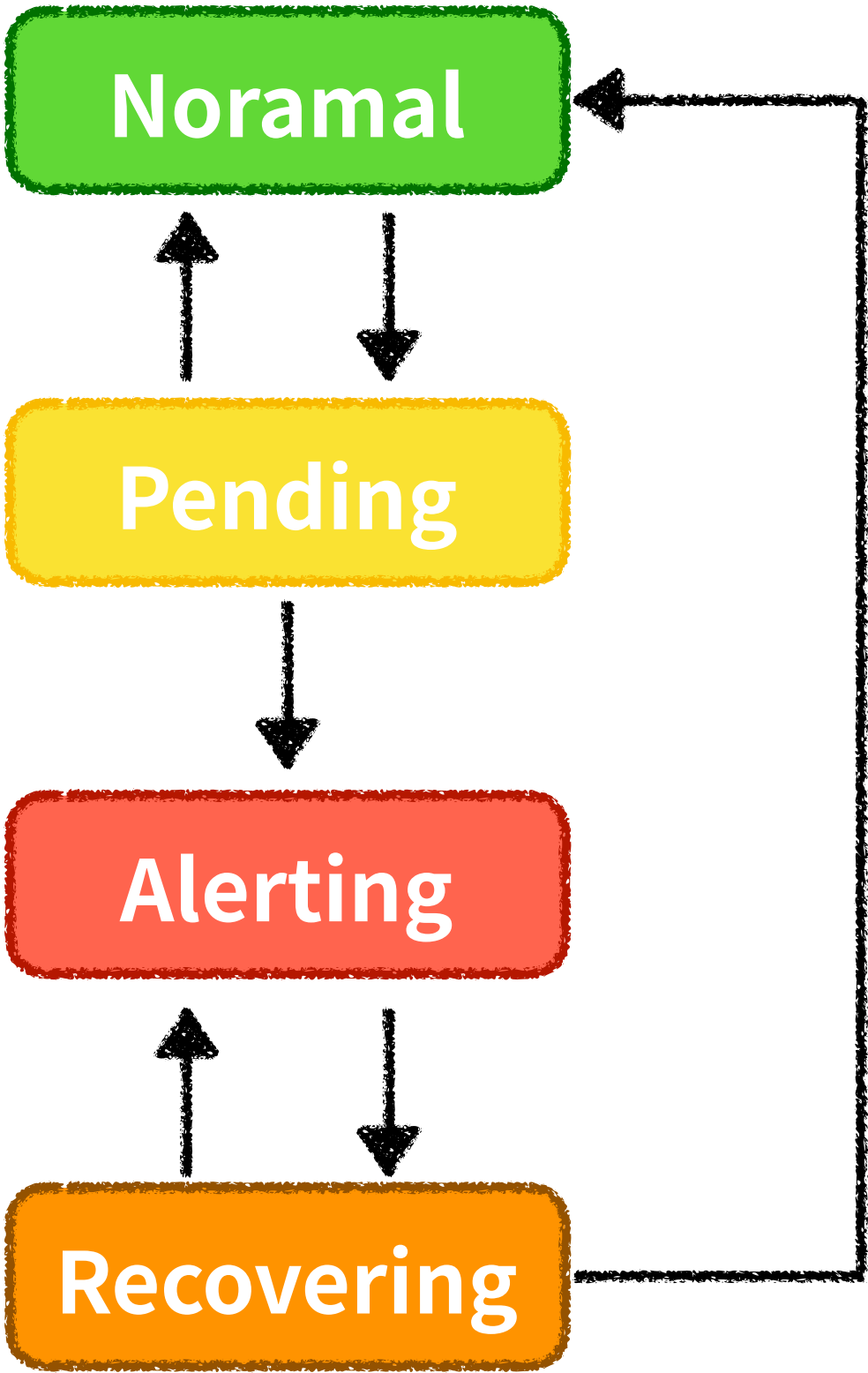
● Pending Period：10 min

● Recovering Period：10 min

Time	CPU 使用率	State	持續時間
00:30	80%	● Normal	
00:35	91%	● Pending	0 min
00:40	93%	● Pending	5 min
00:45	92%	● Alerting	10 min
00:50	91%	● Alerting	
00:55	88%	● Recovering	0 min
01:00	85%	● Recovering	5 min
01:05	80%	● Normal	10 min

← 滿足 Pending

← 滿足 Recovering



Alert State 轉換 Flow

Alerting - Alert Rule

告警條件：CPU 使用率 > 90%

查詢週期：5 min

● Pending Period：10 min

● Recovering Period：10 min

Time	CPU 使用率	State	持續時間
00:30	80%	● Normal	
00:35	91%	● Pending	0 min
00:40	93%	● Pending	5 min
00:45	92%	● Alerting	10 min
00:50	91%	● Alerting	
00:55	88%	● Recovering	0 min
01:00	85%	● Recovering	5 min
01:05	80%	● Normal	10 min

設定查詢語法與告警條件

2. Define query and alert condition

Define query and alert condition ? Need help?

Prometheus

Options

10 minutes

Kick start your query

Explain

Run queries

Builder

Code

Metrics browser >

sum(rate(container_cpu_usage_seconds_total{name=~"app.*"}[5m])) by (name) *100

>

Options

Legend: Auto

Format: Time series

Step: auto

Type: Instant

Table

{name="app-a"}

1.35468

{name="app-b"}

1.2824

{name="app-c"}

93.26604

{name="app-d"}

91.63832

Alert condition

WHEN QUERY

IS ABOVE

90

{name="app-a"}

0

Normal

{name="app-b"}

0

Normal

{name="app-c"}

1

Firing

{name="app-d"}

1

Firing

Preview alert rule condition

Alerting - Alert Rule

告警條件：CPU 使用率 > 90%

查詢週期：5 min

● Pending Period：10 min

● Recovering Period：10 min

Time	CPU 使用率	State	持續時間
00:30	80%	● Normal	
00:35	91%	● Pending	0 min
00:40	93%	● Pending	5 min
00:45	92%	● Alerting	10 min
00:50	91%	● Alerting	
00:55	88%	● Recovering	0 min
01:00	85%	● Recovering	5 min
01:05	80%	● Normal	10 min

設定查詢週期、等待時間

4. Set evaluation behavior

Define how the alert rule is evaluated. [? Need help?](#)

Evaluation group and interval

5m-group

or

+ New evaluation group

All rules in the selected group are evaluated every 5m.

Pending period

Period during which the threshold condition must be met to trigger an alert. Selecting "None" triggers the alert immediately once the condition is met.

10m

None5m10m15m20m25m

Keep firing for

Period during which the alert will continue to show up as firing even though the threshold condition is no longer breached. Selecting "None" means the alert will be back to normal immediately.

10m

None5m10m15m20m25m

> Configure no data and error handling

Alerting - Alert Rule

設定通知方式

5. Configure notifications

Select who should receive a notification when an alert rule fires.

Recipient

Notifications for firing alerts are routed to a selected contact point. [Need help?](#)

Alertmanager: grafana

Contact point

webhook

[View or create contact points](#)

Webhook <http://webhook:8000/webhook>

Muting, grouping and timings (optional)

Contact points

Choose how to notify your contact points when an alert instance fires

Alertmanager Grafana

[Manage permissions](#)

Update contact point

Name *

webhook

Integration

Webhook

URL

[Test](#) [Duplicate](#) [Delete](#)

[http://webhook:8000/webhook](#)

> Optional Webhook settings

> Notification settings

+ Add contact point integration

[Save contact point](#) [Cancel](#)

Alerting - Alert Rule

設定通知額外資訊

6. Configure notification message
Add more context to your alert notifications. ⓘ [Need help?](#)

Summary (optional)
Short summary of what happened and why.

P0 Emergency

Description (optional)
Description of what the alert rule does.

Yo, we got a huge problem!

Runbook URL (optional)
Webpage where you keep your runbook for the alert.

https://

Custom annotation name and content

note

Better call Saul!

+ Add custom annotation 🌐 Link dashboard and panel

額外資訊效果

```
> 2025-06-04 14:28:58,505 INFO [root] [webhook.py:24] - {
  "receiver": "test",
  "status": "firing",
  "alerts": [
    {
      "status": "firing",
      "labels": {
        "alertname": "TestAlert",
        "grafana_folder": "Test Folder",
        "instance": "Grafana"
      },
      "annotations": {
        "description": "Yo, we got a huge problem!",
        "note": "Better call Saul!",
        "summary": "P0 Emergency"
      },
      "startsAt": "2025-06-04T14:28:58.481649676Z",
      "endsAt": "0001-01-01T00:00:00Z",
```


Alerting - Alert Rule View

檢視 Alert Rule 詳細資訊

Home > Alerting > Alert rules

Search...

Alert rules

Rules that determine whether an alert will fire

Search by data sources ⓘ

All data sources

Dashboard

Select dashboard

State

FiringNormalPendingRecovering

Rule type

AlertRecording

Health

OkNo DataError

Contact point

Choose

Search ⓘ

Search

View as

GroupedListState

1 rule1 pending

Grafana-managed

Provision Dashboards > 1m-group

1 pending1mProvisioned

State	Name	Health	Summary	Next evaluation	Actions
> Pending for 22s	Provision Alert - Request Rate	Provisioned	ok	Request Rate Alert	in a few secondsViewMore

Home > Alerting > Provision Dash... > 1m-group > Provision Alert - R...

Search...

Provision Alert - Request Rate

Pending

Evaluation interval

Every 1m

More

Request Rate Alert

Query and conditions

Instances3

History

Details

Versions

A

Prometheus

16m to now

View in Explore

rate(fastapi_requests_total{}[3m])

Table

{app_name="app-a", instance="app-a:8000", job="app-a", method="GET", path="/metrics"}	0.2
{app_name="app-b", instance="app-b:8000", job="app-b", method="GET", path="/metrics"}	0.2
{app_name="app-c", instance="app-c:8000", job="app-c", method="GET", path="/metrics"}	0.2

C

Threshold

Alert condition

Input

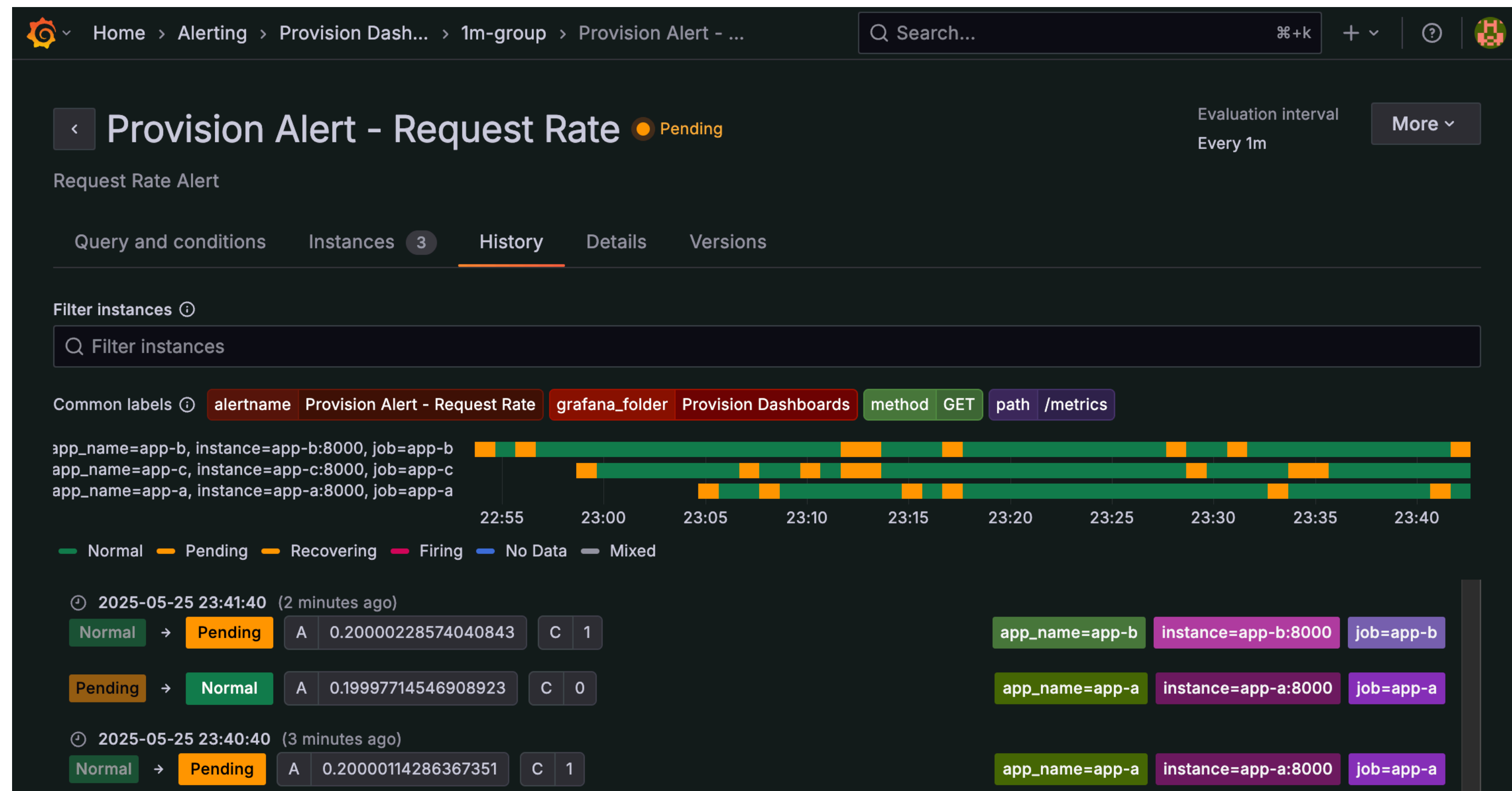
A

Is above 0.2

{app_name="app-a", instance="app-a:8000", job="app-a", method="GET", path="/metrics"}	0	Normal
{app_name="app-b", instance="app-b:8000", job="app-b", method="GET", path="/metrics"}	1	Firing
{app_name="app-c", instance="app-c:8000", job="app-c", method="GET", path="/metrics"}	0	Normal

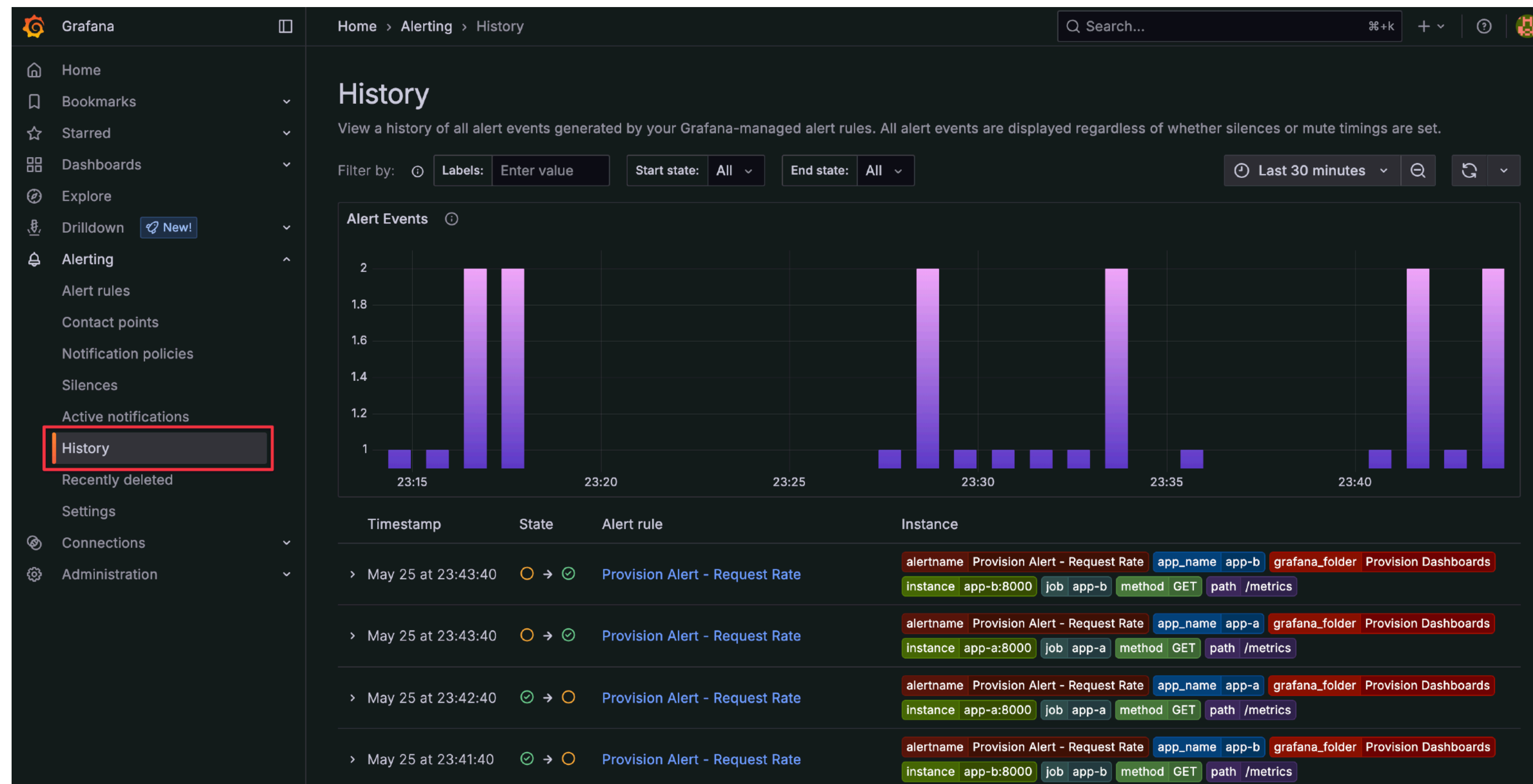
Alerting - Alert Rule View

- Alert State History：Grafana 10 新功能，啟用後搭配 Loki 能以視覺化的方式查看 Alert Rule 中每個 Instance 的 State 變化紀錄



Alerting - Alerting History

- Alerting History Page：Grafana 11.2 新功能，啟用後搭配 Loki 可以查看所有 Alert 的 State 變化紀錄



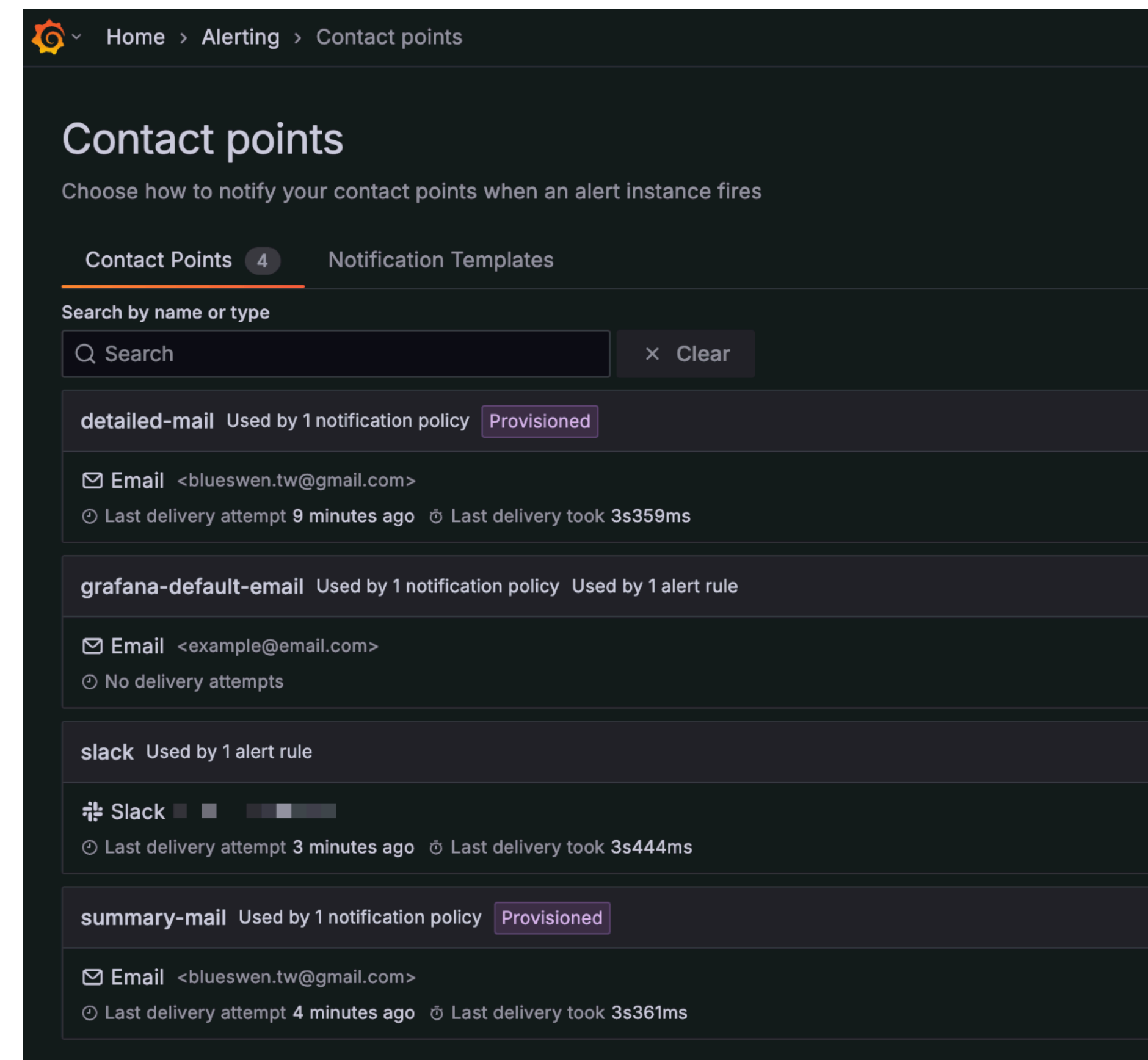
Alerting - Contact Point

Grafana Alerting 支援多種通知方式（Contact Point）

Supported contact point integrations

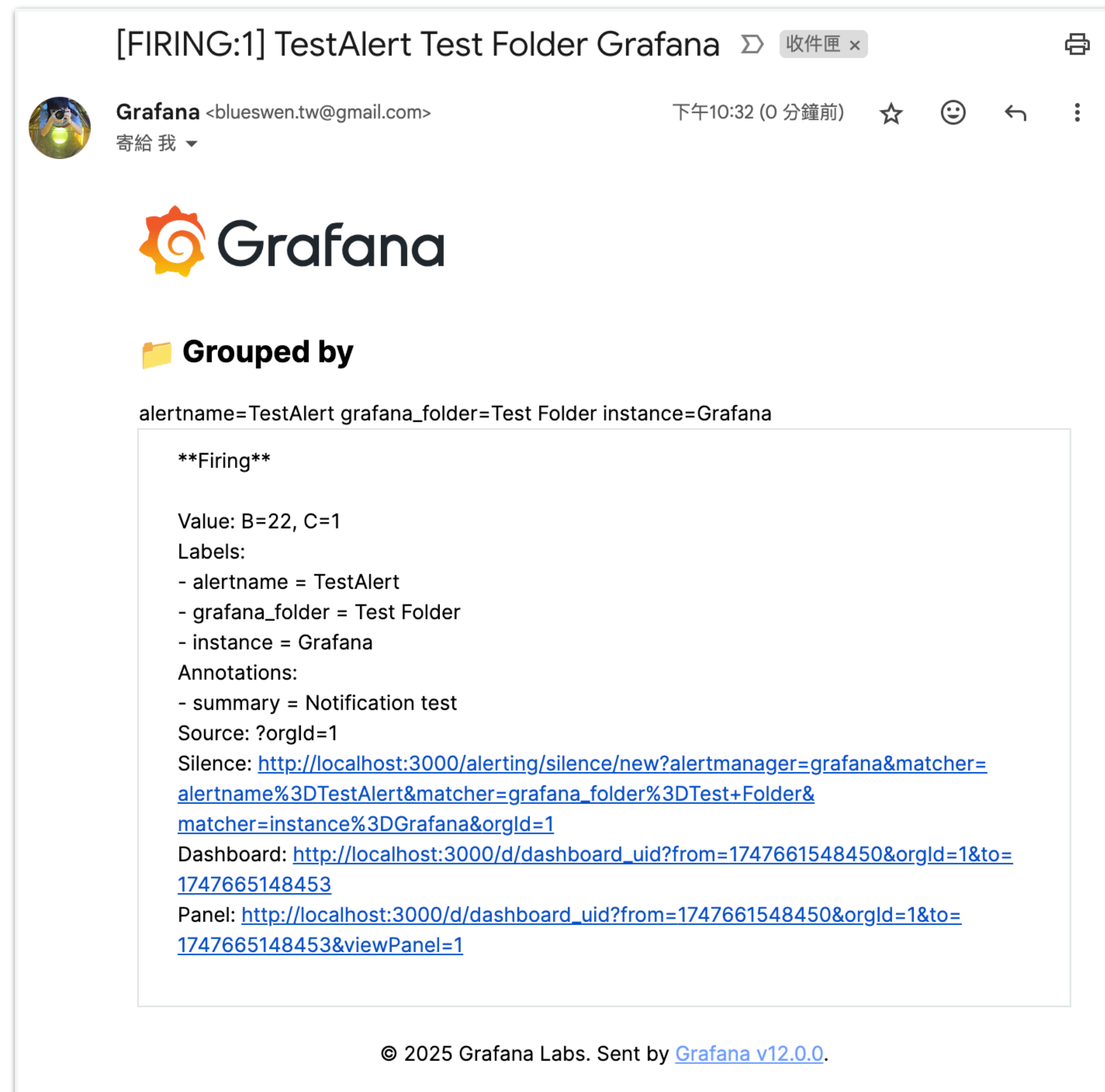
Each contact point integration has its own configuration options and setup process. The following list shows the contact point integrations supported by Grafana.

- Alertmanager
- [AWS SNS](#)
- Cisco Webex Teams
- DingDing
- [Discord](#)
- [Email](#)
- [Google Chat](#)
- [Grafana IRM](#)
- Kafka REST Proxy
- [Jira](#)
- ~~Line~~
- [Microsoft Teams](#)
- [MQTT](#)
- [Opsgenie](#)
- [PagerDuty](#)
- Pushover
- Sensu Go
- [Slack](#)
- [Telegram](#)
- Threema Gateway
- VictorOps
- [Webhook](#)
- WeCom

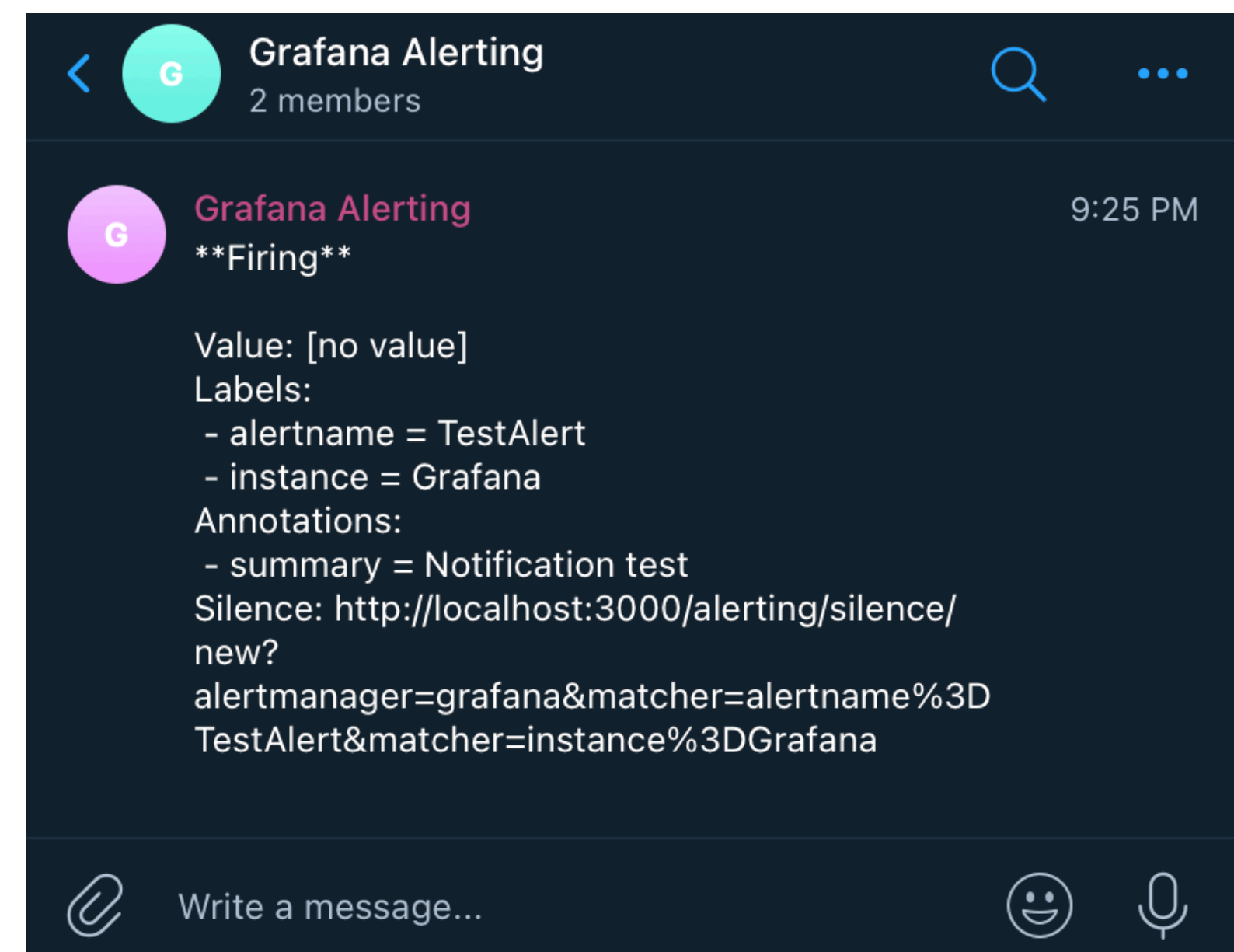
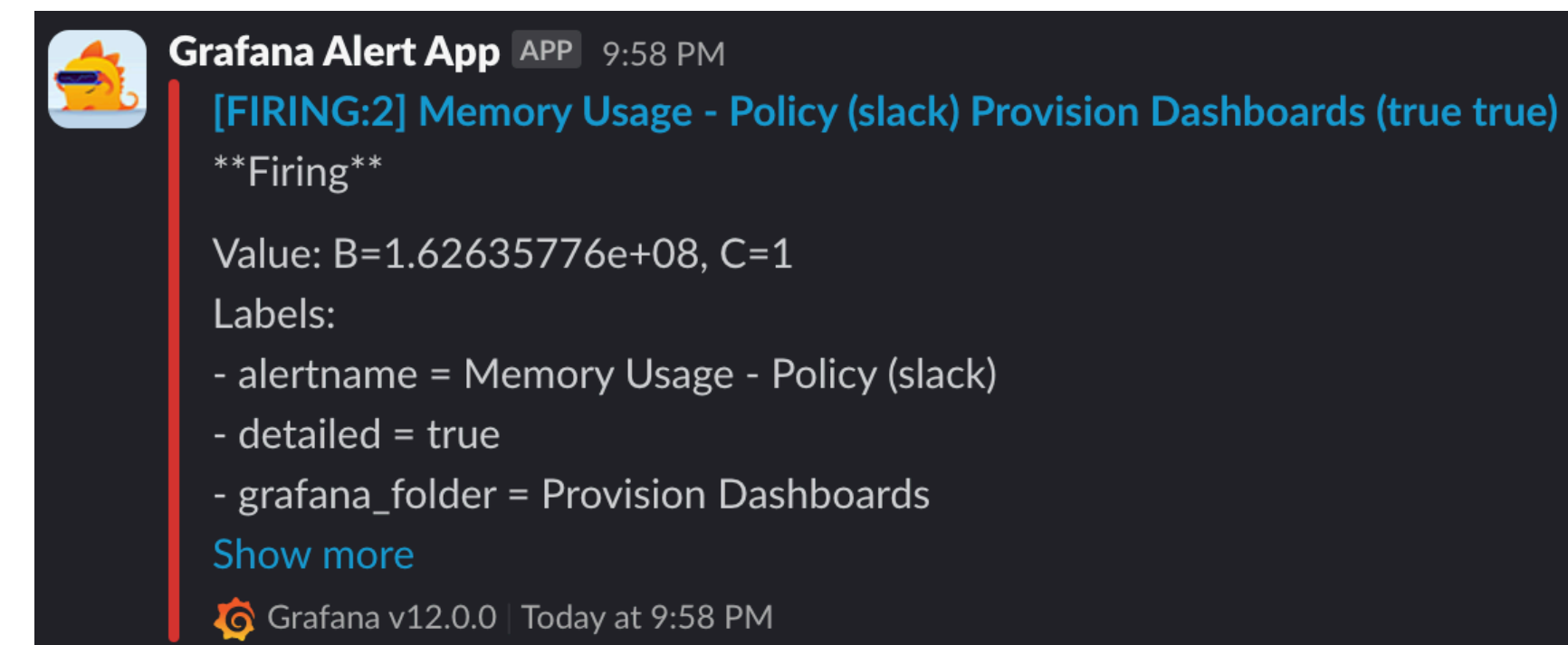


*Line Notify 已於 2025 年 3 月 31 日結束服務

Alerting - Contact Point : Email 、 Slack 、 Telegram

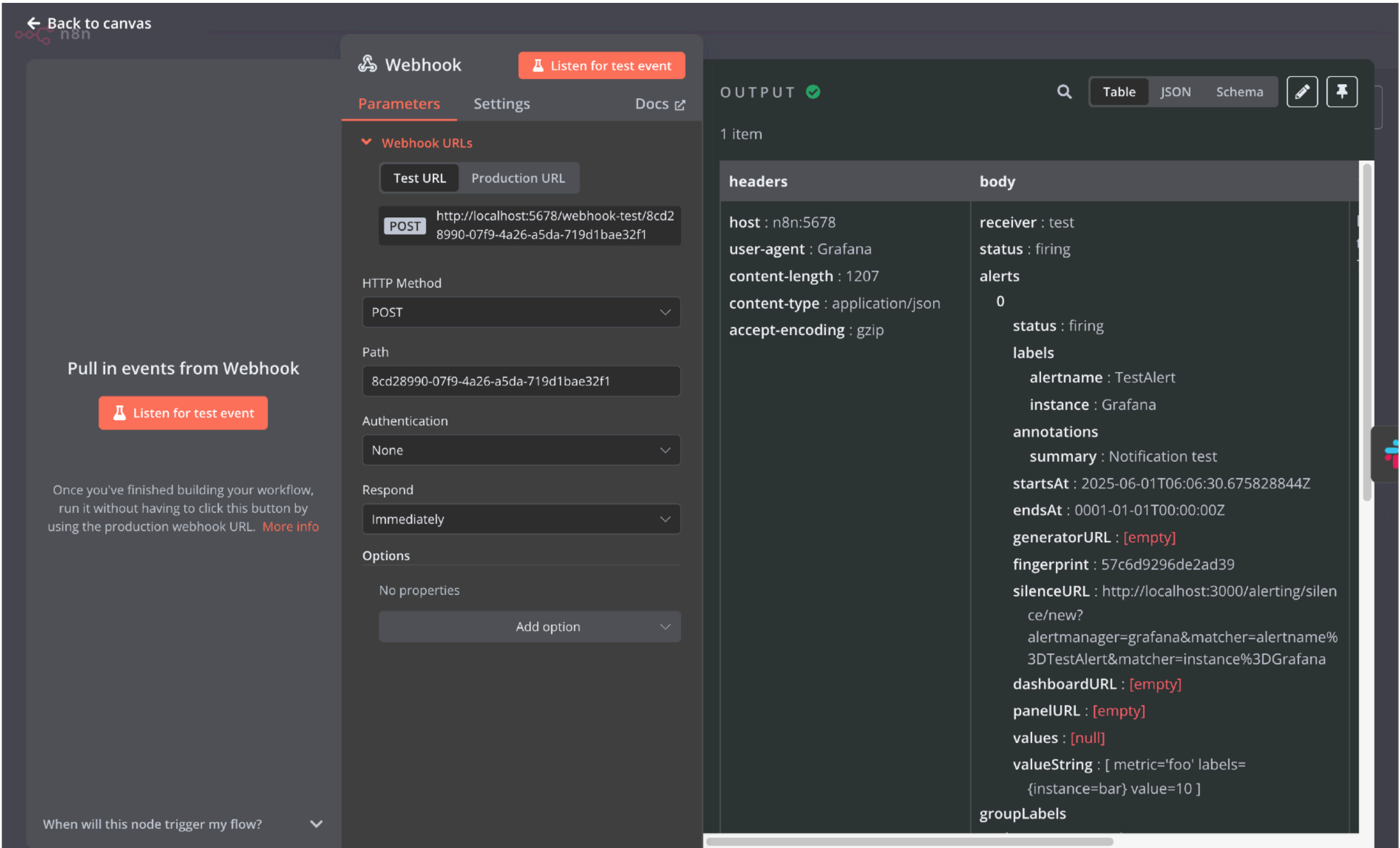


Grafana Zero to Hero - Alerting：強化告警通知

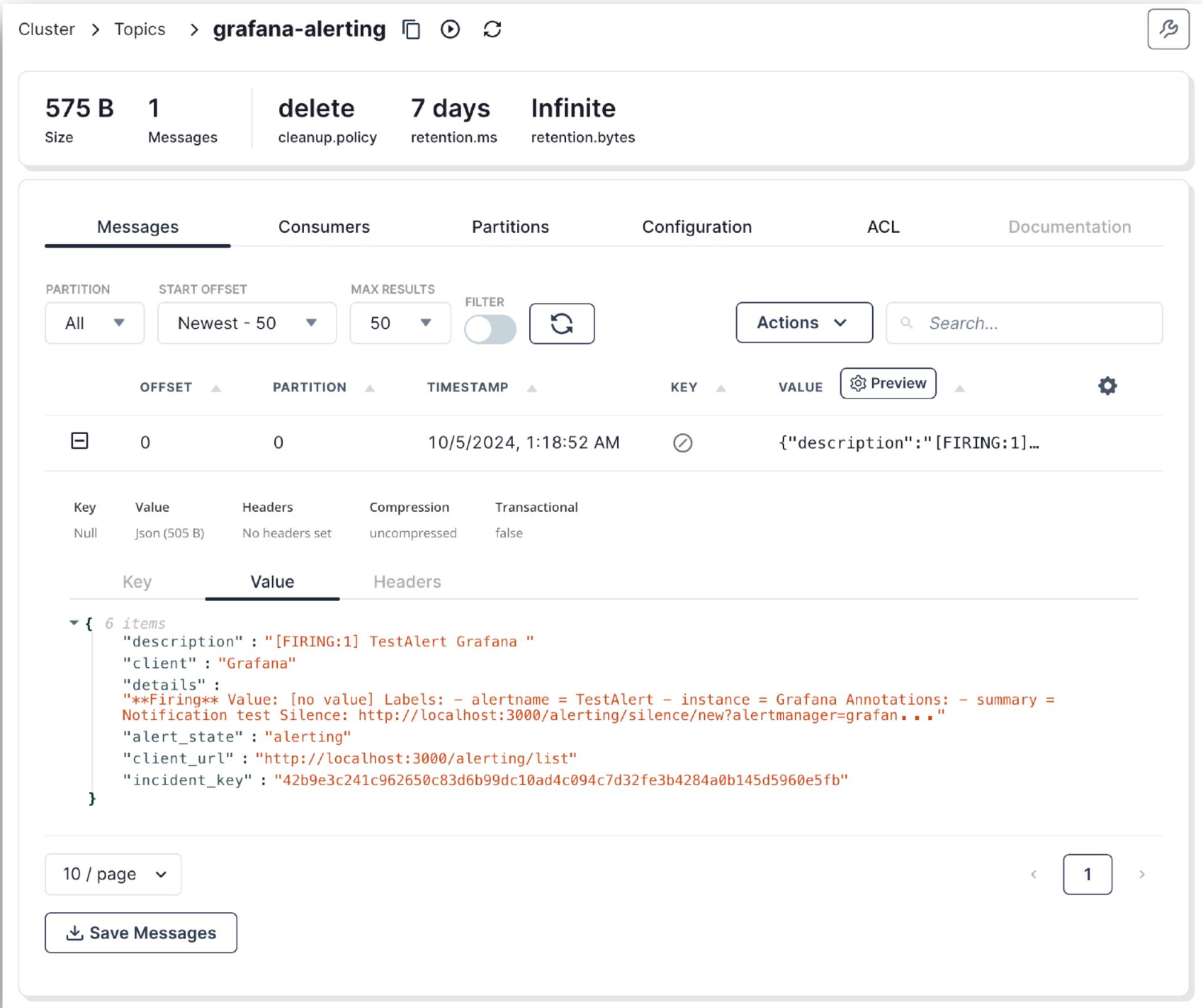


Grafana Zero to Hero - Notification：隨時隨地接收告警

Alerting - Contact Point : Event - Webhook 、 Kafka



Webhook with n8n



Kafka

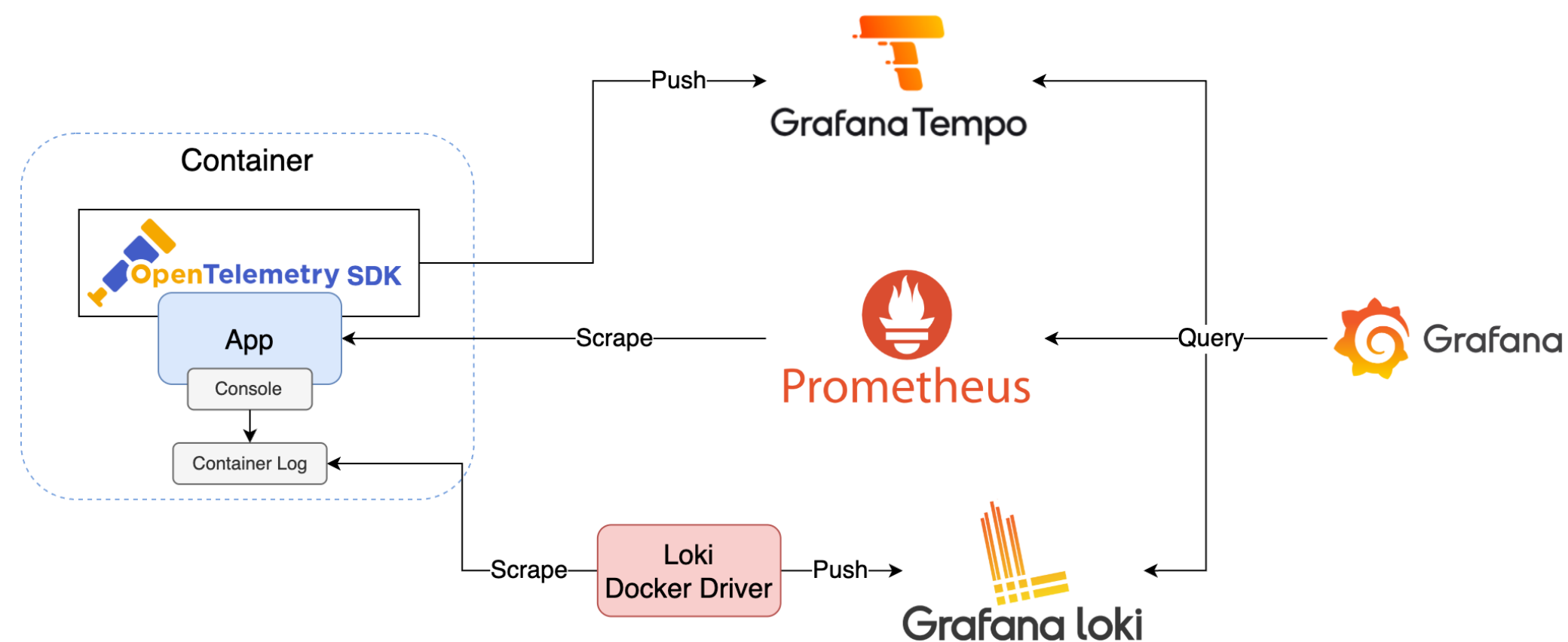
Hands-on - Lab 02: Grafana Alerting

- GitHub URL : [Lab 02](#)
- Goals : 設定與檢視 Grafana Alerting
- Task
 - 建立 Container Memory 使用量超過 100 MB 的 Alert
 - 每 30 秒查詢一次
 - 連續 1 分鐘都高於觸發條件，進入 Alerting 狀態
 - 連續 3 分鐘都低於觸發條件，解除 Alerting 狀態
 - Contact Point : 使用預先建立的 webhook Contact Point

Recap

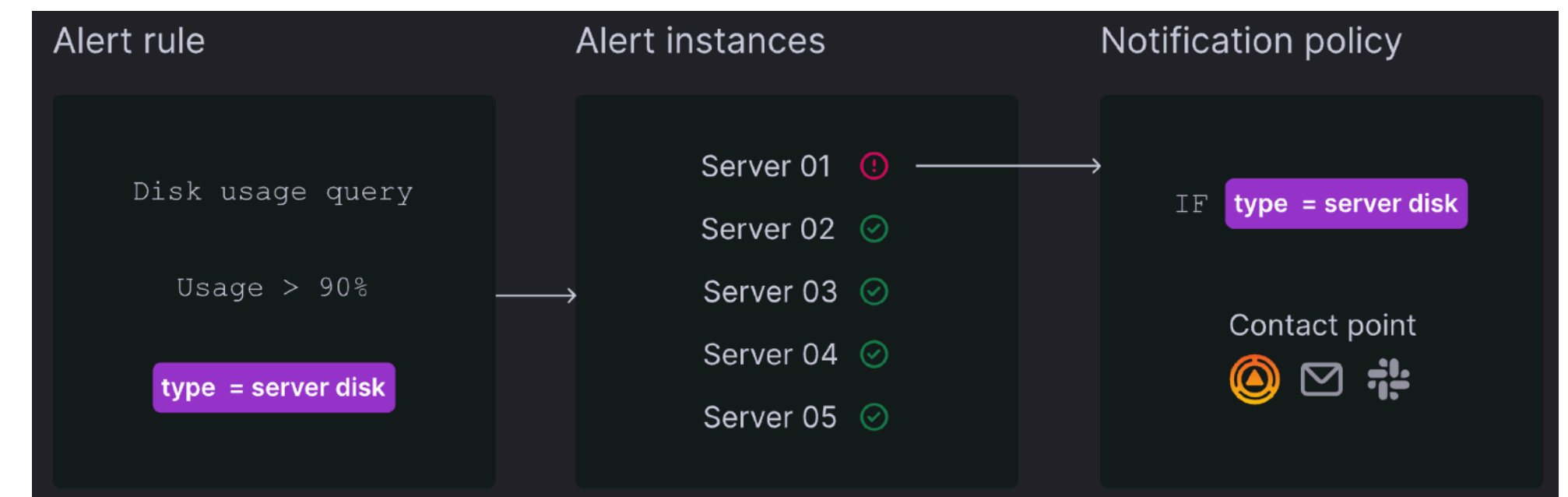
Observability

- 透過各種**資訊**，**清楚**了解系統狀態
- 資訊的**生成**、**收集**、**儲存**、**使用**
- 打破 Silo，建立**關聯**，產生**綜效**



Alerting

- 定期執行**查詢**，根據**條件**轉換**狀態**
- 支援多種**訊息**、**事件**通知機制



延伸閱讀

- Demo Project
 - [FastAPI Observability](#)
 - [Spring Boot Observability](#)
 - [FastAPI Tracing with Jaeger through OpenTelemetry](#)
 - [OpenTelemetry Application Performance Management](#)
- [時光之鏡：透視過去、現在與未來的 Observability：概念、工具與應用](#)
- [Grafana Zero to Hero](#)：Grafana 基礎操作與進階應用
- [Observability 101：從零開始了解可觀測性：可觀測性基礎介紹](#)
- [全方位強化 Python 服務可觀測性：以 FastAPI 和 Grafana Stack 為例](#)

更多 Observability 議程

- 2025-06-05
 - 為什麼我們需要 Observability ?
 - Observability 入門班：可觀測性的核心技術架構與 OpenTelemetry 實作指南
 - SRE的未來：LLM 引領的可觀測性與可追溯性革命
 - 從開發到架構設計的可觀測性實踐
 - Data Observability：企業資料管理技術的未來顯學
 - 深入 Grafana 探討告警事件中心實踐
 - 為什麼你裝了一堆 O11y 工具，卻沒人用？
- 2025-06-06
 - 【Bootcamp】從 Day 0 開始的可觀測性：用 ODD 與 SLO 的實作工作坊
 - 【Bootcamp】初探 LLM 可觀測性：從概念到實踐
 - 更高效率低成本的 Observability 2.0 時代即將來臨
 - 黑暗大陸內的 Containerization & Pulumi IaC

工商時間



- 徵 Backend Engineer
- Tech Stack
 - AWS
 - FastAPI、Django
 - DBT、Celery
 - PostgreSQL、MongoDB、Redis
 - Kubernetes、Docker

Job Description



全面介紹 Grafana、Prometheus、OpenTelemetry !

可觀測性入門指南

Metrics、Logs、Traces 三大實戰應用，
用 24 個 Lab 鞏固你的 SRE 技能樹

劉義璋 (Blueswen) 著



實現全面系統監控和分析！

從零開始，構建強大的可觀測性架構

概念導覽
深入淺出解析
可觀測性概念

完整範例
使用 Docker Compose
展示多種實戰範例

多處適用
Kubernetes 與
傳統服務皆適用

資訊整合
結合 Metrics、Logs
和 Traces 全面洞察



iThome
鐵人賽

天瓏網路書店



可觀測性概念、工具介紹