OpenTelemetry : Drupal sous haute surveillance

Par Tarek DJEBALI

Tarek DJEBALI

Symdrik

Consultant / Expert Drupal

D.O: tarekdj

X: _tarekdj

Linkedin: in/tarekdj

Symdrik

www.symdrik.com

Agence web & mobile fondée en 2008

80 collaborateurs

Paris / Tunis

Parmis nos clients :

ACCOR

SNCF

TotalEnergies

france tv

Cultura

La culture avec un grand AAAH!

meetic

OVHcloud

OVHcloud

https://www.ovhcloud.com/fr/

L'observabilité: un enjeu majeur!

- C.A. de 897 millions d'euros pour l'année 2023*
- Vente de produits pouvant atteindre plusieurs milliers d'euro.
- Le web comme canal de vente

** Source : https://corporate.ovhcloud.com/fr/newsroom/news/fy2023-annual-results/*

SOMMAIRE

1. Introduction à l'observabilité
2. OpenTelemetry
3. OpenTelemetry & Drupal
4. Démo

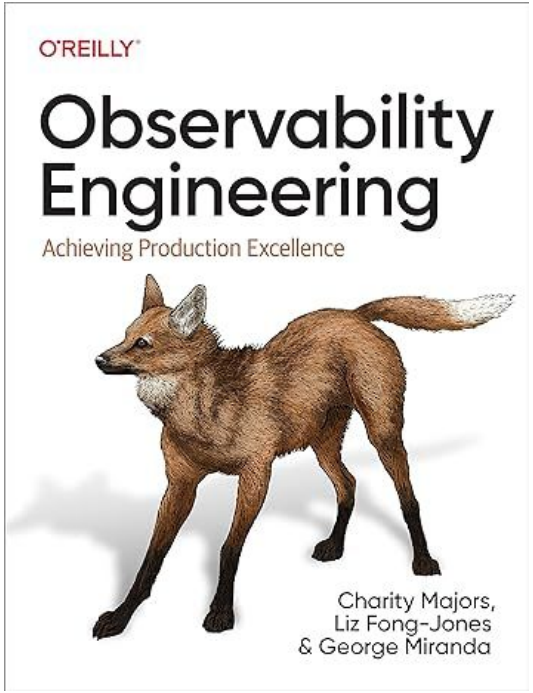
Introduction à l'observabilité

Définition mathématique

Le terme « observabilité » a été inventé par l'ingénieur Rudolf E. Kálmán en 1960.

Dans la théorie du contrôle, l'observabilité est définie comme une façon pour déterminer le bon fonctionnement interne d'un système à travers la connaissance de ses sorties externes.

Source: O'Reilly - Observability Engineering



L’observabilité des systèmes informatiques

Un système est dit observable s’il permet d’analyser son fonctionnement interne à travers ses signaux externes.

https://unsplash.com/fr/@claudegabriel

Les 3 piliers (principaux) de l’observabilité

https://imgflip.com/i/8hl8uq

Les logs

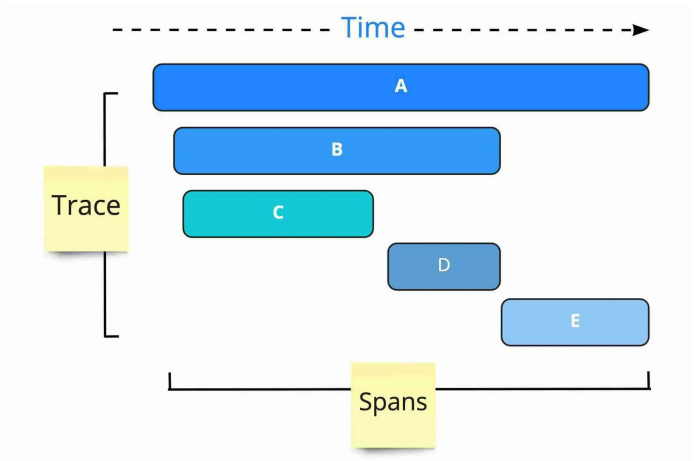
Un log est un enregistrement texte horodaté, structuré ou non structuré, avec des métadonnées décrivant un événement qui s'est produit dans le système

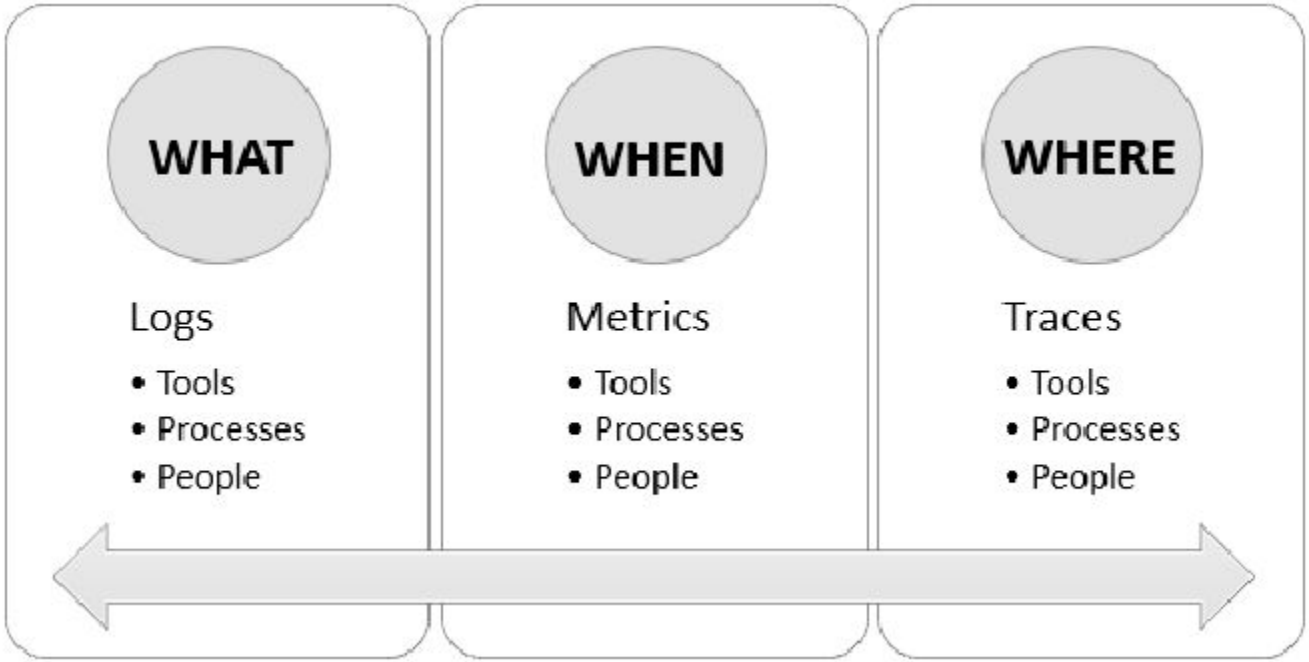
Les metrics

Une mesure capturée au moment de l'exécution du programme (nombre de requêtes, temps d'exécution ...)

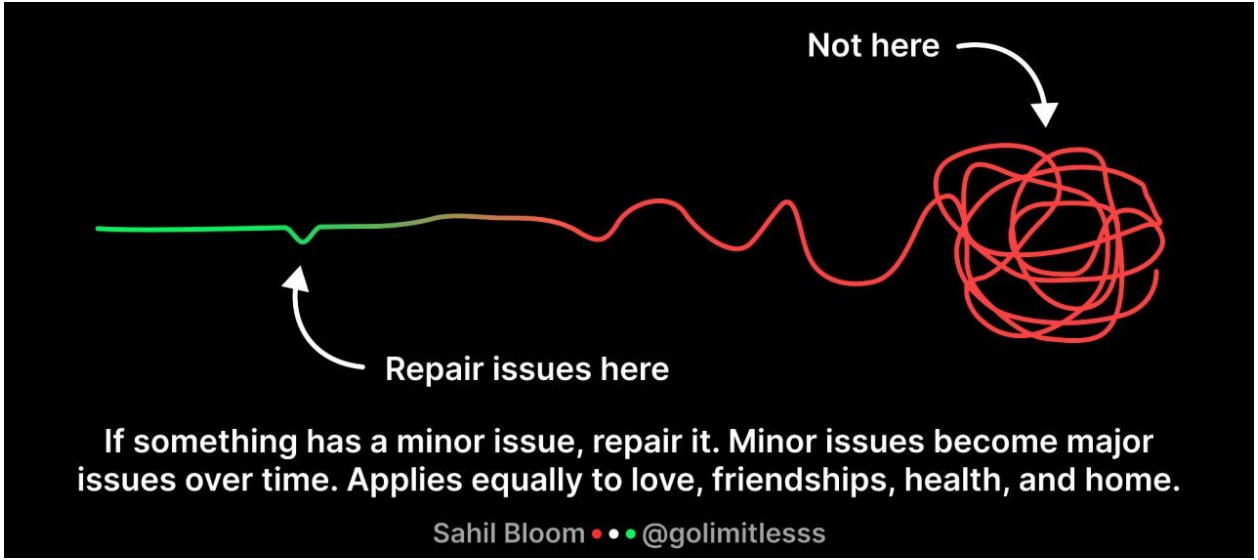
Les traces

Les traces nous donnent une vue d'ensemble de ce qui se passe lorsqu'une requête est adressée à une application.




WHAT
WHEN
WHERE
Logs
• Tools
• Processes
• People
Metrics
• Tools
• Processes
• People
Traces
• Tools
• Processes
• People

Pourquoi?


Not here
Repair issues here
If something has a minor issue, repair it. Minor issues become major issues over time. Applies equally to love, friendships, health, and home.
Sahil Bloom @golimitlesss

OpenTelemetry (OTel)

C’est quoi?

OpenTelemetry est standard + un ensemble d’outils libres conçus pour implémenter l’observabilité des applications.

Il est le résultat de la fusion d’OpenTracing avec Opencesus.

Langages supportés

https://opentelemetry.io/docs/languages/

Language	Traces	Metrics	Logs
C++	Stable	Stable	Stable
C#/.NET	Stable	Stable	Stable
Erlang/Elixir	Stable	Experimental	Experimental
Go	Stable	Stable	In development
Java	Stable	Stable	Stable
JavaScript	Stable	Stable	Experimental
PHP	Stable	Stable	Stable
Python	Stable	Stable	Experimental
Ruby	Stable	In development	In development
Rust	Beta	Alpha	Alpha
Swift	Stable	Experimental	In development

Instrumentation

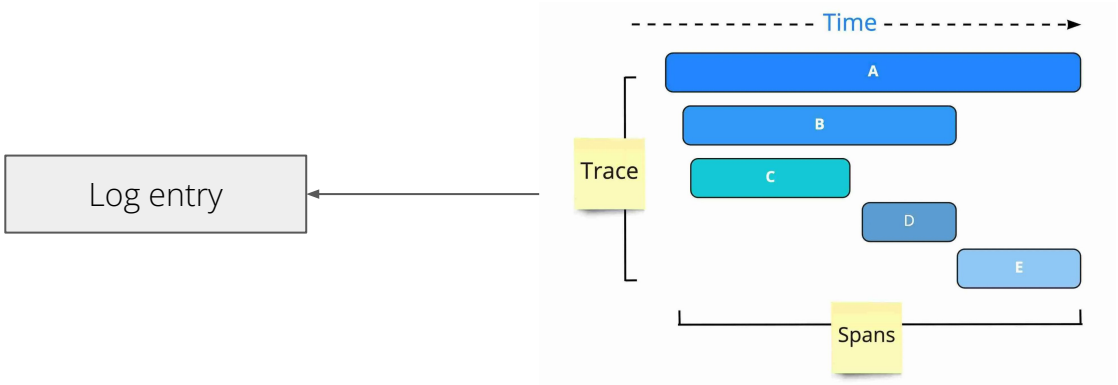
“L’instrumention” désigne le processus de collecte des données:

- Code-based via API/SDK
- Zero-code a.k.a auto-instrumentation

Propagation de context

Le context est un objet contenant des informations passé du service appelant au service appelé.

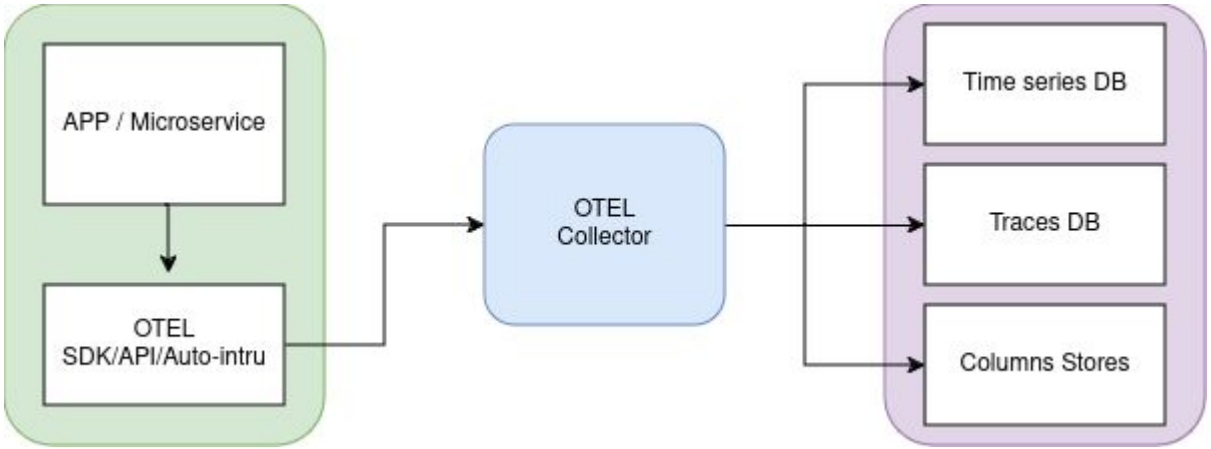
Grâce à ce mécanisme, les signaux peuvent être corrélés les uns aux autres, quel que soit l'endroit où ils sont générés.



Les composants d’OTel

- Spécifications (API/SDK, OTLP, Collector)
- Collector
- API/SDK spécifique à chaque language
- Opérateurs K8s
- FaaS assets (serverless artifacts)

Topologie (très très simplifiée)



OTel Collector

Configuration

```
receivers:
 otlp:
   protocols:
     grpc:
       endpoint: 0.0.0.0:4317
     http:
       endpoint: 0.0.0.0:4318
processors:
 batch:

exporters:
 otlp:
   endpoint: otelcol:4317

extensions:
 health_check:
 pprof:
 zpages:
```

```
service:
 extensions: [health_check, pprof, zpages]
 pipelines:
   traces:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   metrics:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   logs:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
```

Configuration - Receivers

```
receivers:
 otlp:
   protocols:
     grpc:
       endpoint: 0.0.0.0:4317
     http:
       endpoint: 0.0.0.0:4318
processors:
 batch:

exporters:
 otlp:
   endpoint: otelcol:4317

extensions:
 health_check:
 pprof:
 zpages:
```

```
service:
 extensions: [health_check, pprof, zpages]
 pipelines:
   traces:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   metrics:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   logs:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
```

Configuration - Processors

```
receivers:
 otlp:
   protocols:
     grpc:
       endpoint: 0.0.0.0:4317
     http:
       endpoint: 0.0.0.0:4318
processors:
 batch:

exporters:
 otlp:
   endpoint: otelcol:4317

extensions:
 health_check:
 pprof:
 zpages:
```

```
service:
 extensions: [health_check, pprof, zpages]
 pipelines:
   traces:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   metrics:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   logs:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
```

Configuration - Exporters

```
receivers:
 otlp:
   protocols:
     grpc:
       endpoint: 0.0.0.0:4317
     http:
       endpoint: 0.0.0.0:4318
processors:
 batch:

exporters:
 otlp:
   endpoint: otelcol:4317

extensions:
 health_check:
 pprof:
 zpages:
```

```
service:
 extensions: [health_check, pprof, zpages]
 pipelines:
   traces:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   metrics:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   logs:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
```

Configuration - Extensions

```
receivers:
 otlp:
   protocols:
     grpc:
       endpoint: 0.0.0.0:4317
     http:
       endpoint: 0.0.0.0:4318
processors:
 batch:

exporters:
 otlp:
   endpoint: otelcol:4317

extensions:
 health_check:
 pprof:
 zpages:
```

```
service:
 extensions: [health_check, pprof, zpages]
 pipelines:
   traces:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   metrics:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   logs:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
```

Configuration - Service

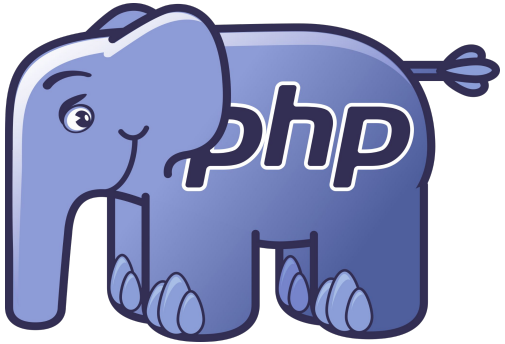
```
receivers:
 otlp:
   protocols:
     grpc:
       endpoint: 0.0.0.0:4317
     http:
       endpoint: 0.0.0.0:4318
processors:
 batch:

exporters:
 otlp:
   endpoint: otelcol:4317

extensions:
 health_check:
 pprof:
 zpages:
```

```
service:
 extensions: [health_check, pprof, zpages]
 pipelines:
   traces:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   metrics:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
   logs:
     receivers: [otlp]
     processors: [batch]
     exporters: [otlp]
```

OTEL & PHP

API/SDK status:

- **Traces:** stable
- **Metrics:** stable
- **Logs:** stable



Registry:
https://opentelemetry.io/ecosystem/registry/?language=php

Documentation:
https://opentelemetry.io/docs/languages/php/

OpenTelemetry & Drupal

Les solutions disponibles

OpenTelemetry	Observability suite
• Version Bêta • Instrumentation manuelle • Basé sur le Events/hooks	• Version Dev • Instrumentation manuelle • Basé sur les Events/hooks • Fortement couplé à Prometheus

Pas d'auto-instrumentation… Pour l'instant 😈

Demo

your high-digital place Open

AXESS
Digital & Business Experts

SMILE

Talan

DRUPALCAMP ROAZHON
Rennes - 28 au 30 mars 2024