



Découvrons ensemble la relève de l'observabilité
avec les logs et traces : Quickwit

BDX/IO à Bordeaux, 08/11/2024

Qui suis-je ?

Idriss Neumann

CEO de comwork.io

SRE/Platform Engineer

Contributeur OSS (incluant les intégrations à l'éco-système CNCF pour Quickwit)



idrissneumann

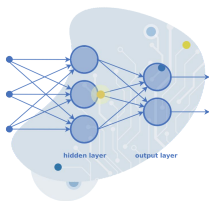
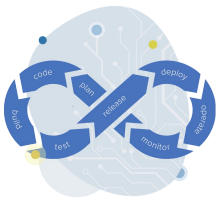


idriss_neumann

Qui sommes nous ?

ESN et éditeur de logiciel basé à Paris et Tunis

4 zone d'expertise: devops & cloud, IOT, full stack dev et AI/ML



The screenshot shows the Comwork dashboard. On the left is a sidebar with navigation links: Dashboard, Projects, Buckets, Registries, Instances, K8s applications, Serverless, Emails, and One Chat. The main area is divided into sections: 'Arguments' with a table, 'Environment variables', 'Callbacks', and a 'Blockly' logic editor.

#	Argument name	Actions
1	name	[edit] [delete]
2	surname	[edit] [delete]

```
Blockly logic:  
Logic  
  handle with: name, surname  
  set an argument with key: name  
  and value: name  
  set an argument with key: surname  
  and value: surname  
  call sync: serverless function  
  with ID: c115c89e-8a8c-4682-bd44-b05e4305ecb  
  and arguments: arguments  
  set result in variable: response  
  set entity to: get value response from key entity  
  set content to: get value entity from key content  
  set result to: get value content from key result  
  return result
```

Site web : comwork.io



Rappel sur l'observabilité

Rappel sur les 3 piliers de l'observabilité

L'**observabilité** est la capacité de mesurer l'état courant d'un système à partir des données qu'il produit qui peuvent être de différentes natures comme les **logs**, les **métriques** et les **traces**.

Logs

Il s'agit d'enregistrements datés et produits par une application afin de fournir des éléments contextuels permettant d'investiguer en cas d'incident

Métriques

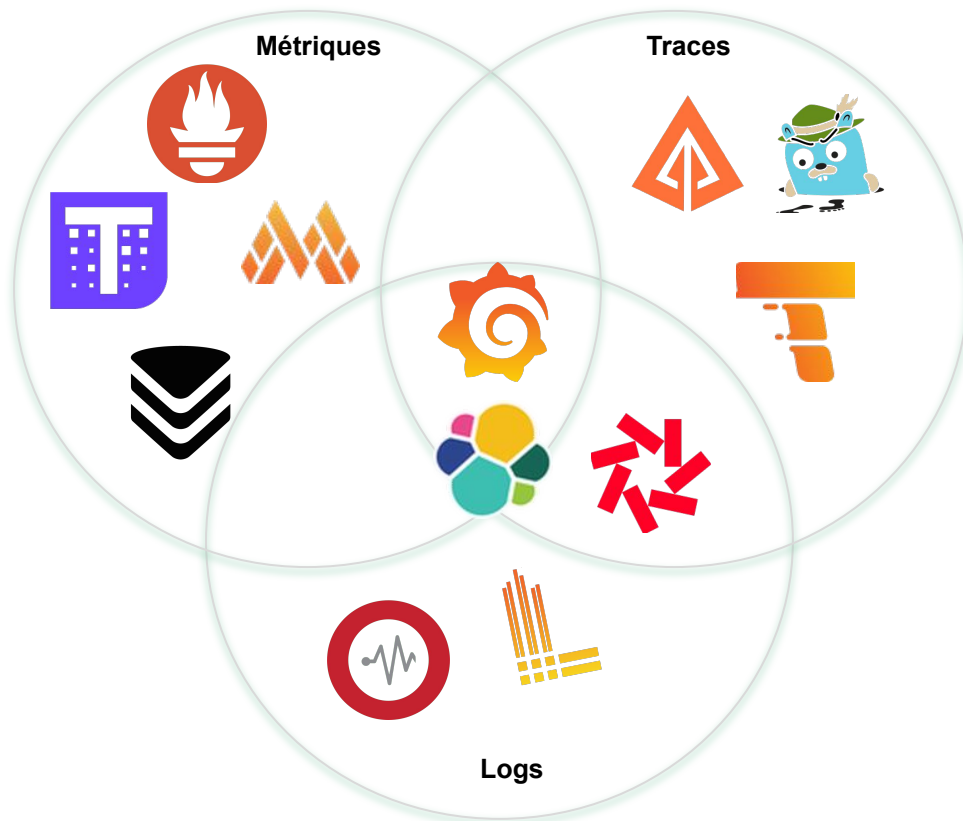
Représentation numérique de données mesurées dans un interval de temps

Traces

Représentation de la relation causal entre plusieurs événements dans un système distribué

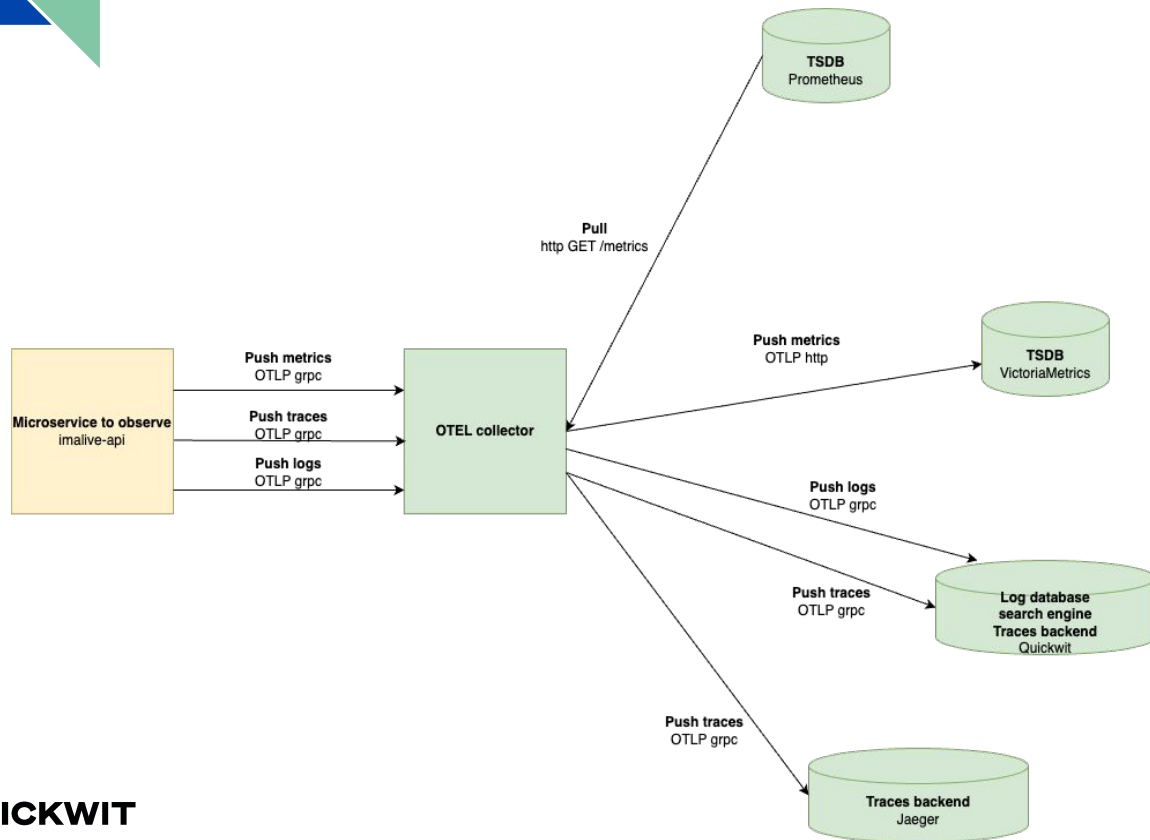
Observability landscape

Classement des outils d'observabilité les plus célèbres



Qu'est-ce qu'OpenTelemetry ?

Un standard d'observabilité interopérable pour les logs, traces et métriques

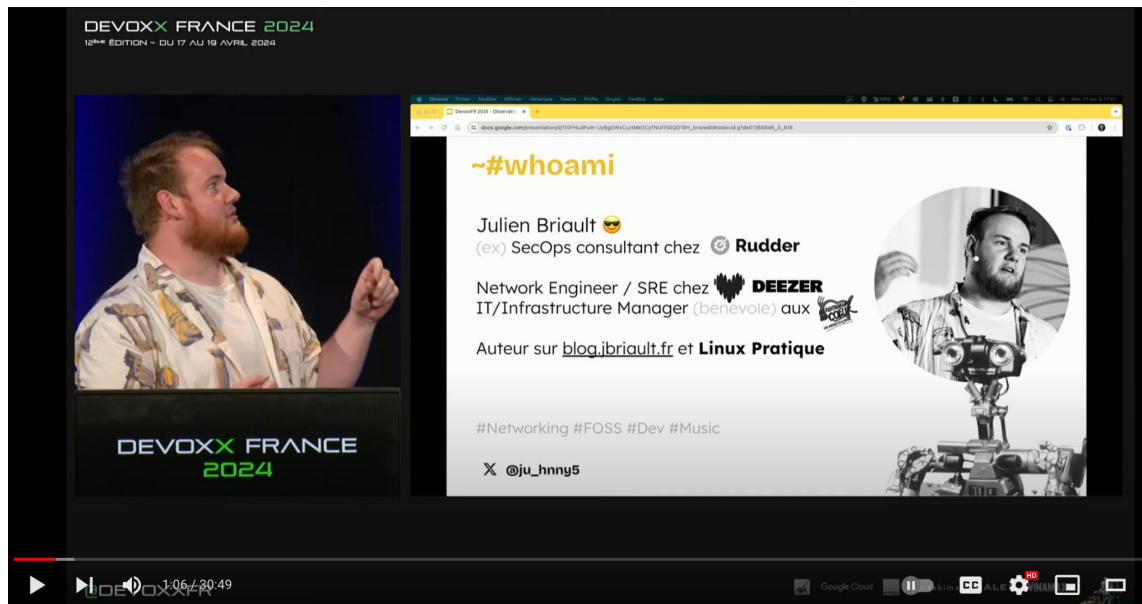


Site web : opentelemetry.io



Qu'est-ce que VictoriaMetrics ?

Petite parenthèse pour aller voir le talk de Julien



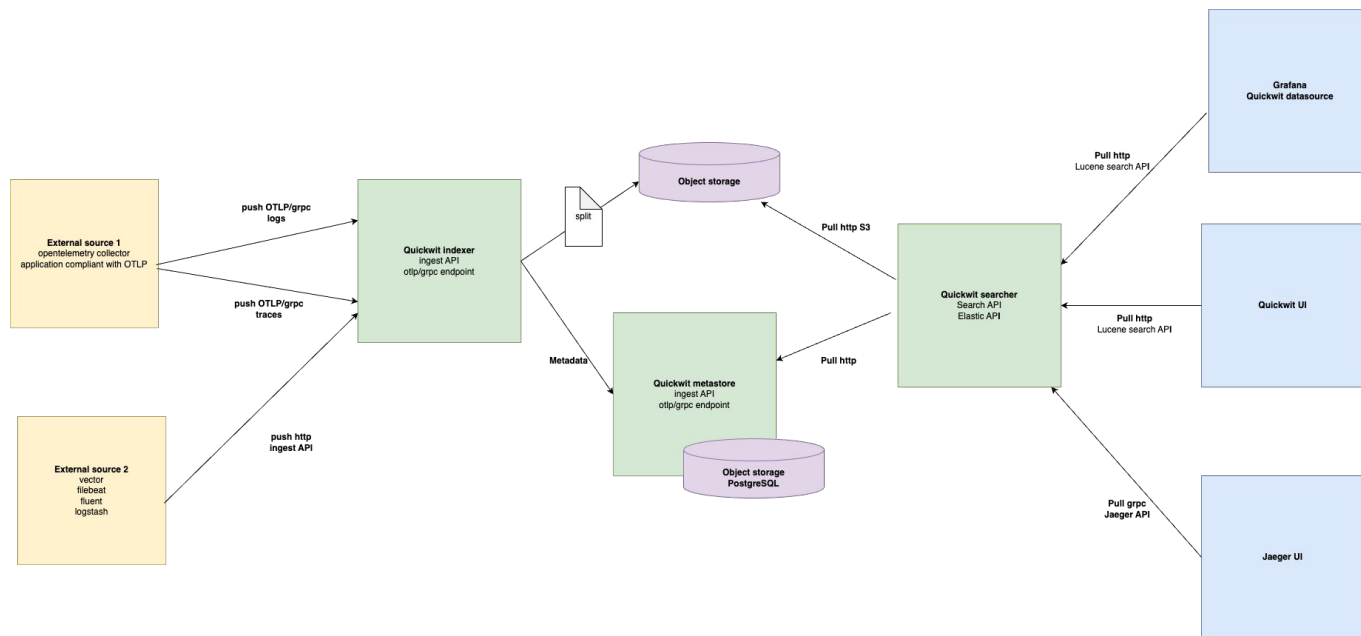
Talk de Julien "Observabilité :
dépoussiérer Prometheus
avec VictoriaMetrics":
youtu.be/bzLfWjUj2k0



Qu'est-ce que Quickwit ?

Solution de moteur de recherche concurrente à Elasticsearch, OpenSearch et Grafana Loki

Un peu le meilleur des deux mondes réunis



Site web : quickwit.io



Pourquoi choisir Quickwit ?

Les raisons de notre choix de cette solution



Comwork Cloud Comwork IOT Our Team

Jobs Training Events **Blog** English

Search Loading...

Recent posts

The Serverless state of art in 2024

Pulumi, the best IaC tool in 2024?

Quickwit, the next generation of modern observability

Docker in production, is it really bad?

Kubernetes or not, that's the question

Quickwit, the next generation of modern observability

September 4, 2024 - 6 min read

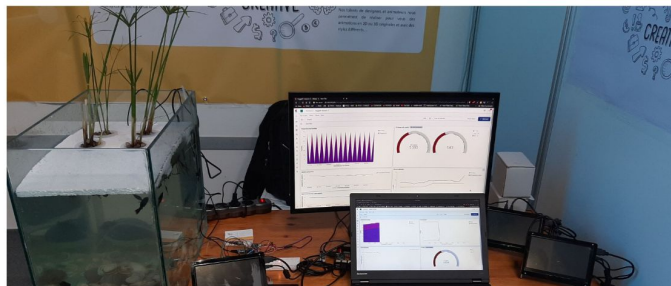


Idriss Neumann
CEO comwork.io

In this blog post, I'll try to explain why we moved from **ElasticStack** to **Quickwit** and **Grafana** and why we choosed it over other solutions.

First, we've been in the observability world for quite some time and have been using ElasticStack for years. I personally used Elasticsearch for more than 10 years and **Apache Solr** before for logging and observability usecases even before Elasticsearch's birth!

We also succeed to use ElasticStack for *IoT (Internet of Things)* projects and rebuilt our own images of Kibana and Elasticsearch for ARM32 and ARM64 before *Elastic* (the company) starts to release official images. We had a lot of fun with it.



Lien : comwork.io/blog/quickwit



Définition des index avec quickwit

Les types

- `text`: chaîne de caractère
- `datetime`: date / timestamp
- `i64`: entier (64 bits)
- `f64`: nombre à virgule flottante (64 bits)
- `u64`: entier non signé (64 bits)
- `ip`: IP address
- `bytes`: valeur binaire ou encodée en base 64
- `json`: objets dynamiques

Les types composites

- `array`: liste de champs
- `object`: nested object

Lien :

quickwit.io/docs/configuration/index-config#doc-mapping



Requêter quickwit

Structure d'une requête

```
field:condition
```

- `field:value: term clause`
- `field:value*: term prefix clause`
- `field:IN [val1 val2 ...]: term set clause`
- `field:"sequence of words": phrase clause`
- `field:"sequence of words*": phrase prefix clause`
- `field:[0 TO 1000]: range clause`
- `*:all`

Lien :

quickwit.io/docs/get-started/query-language-intro



Requêter quickwit

Opérateurs logiques

```
NOT field:condition
```

```
field1:condition1 OR field2:condition2
```

```
field1:condition1 AND field2:condition2
```

Par défaut c'est l'opérateur AND qui s'applique

```
field1:condition1 field2:condition2
```

Vous pouvez grouper et prioriser des prédicats grâce aux parenthèses

```
field1:condition1 AND NOT (field2:condition2 OR field3:condition3)
```

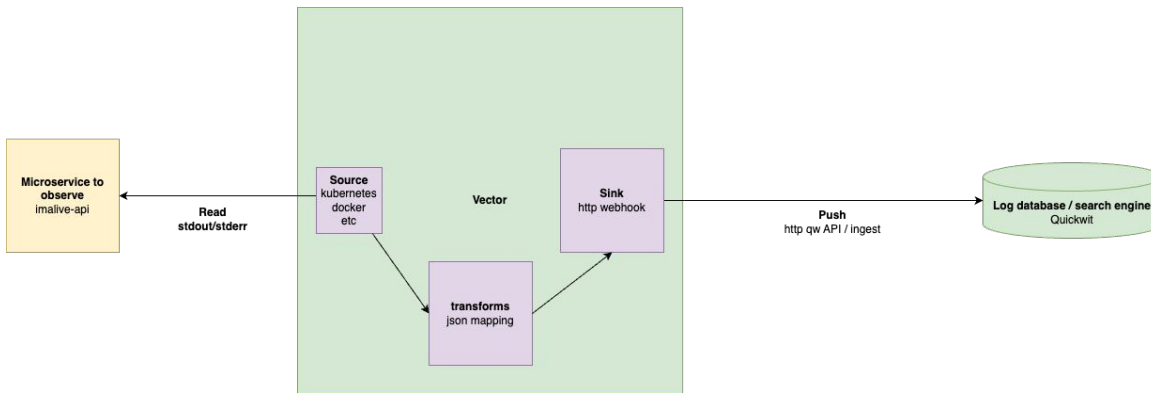
Lien :

quickwit.io/docs/get-started/query-language-intro

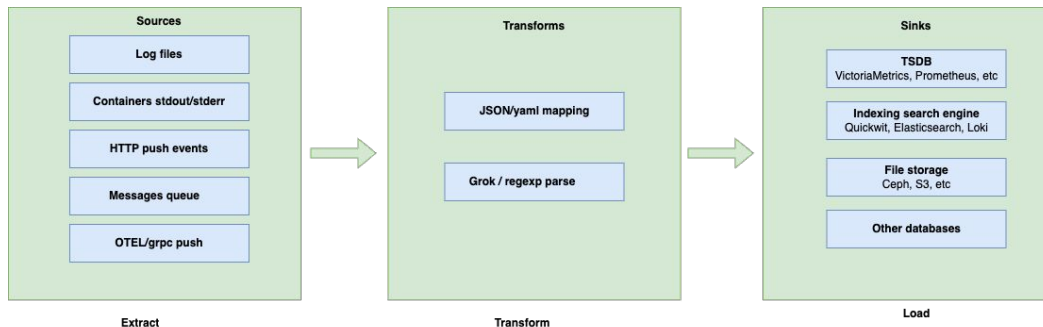


Qu'est-ce que vector ?

Agent de collecte de logs et pipelines d'observabilité / ETL
Très rapide, écrit en Rust par datadog

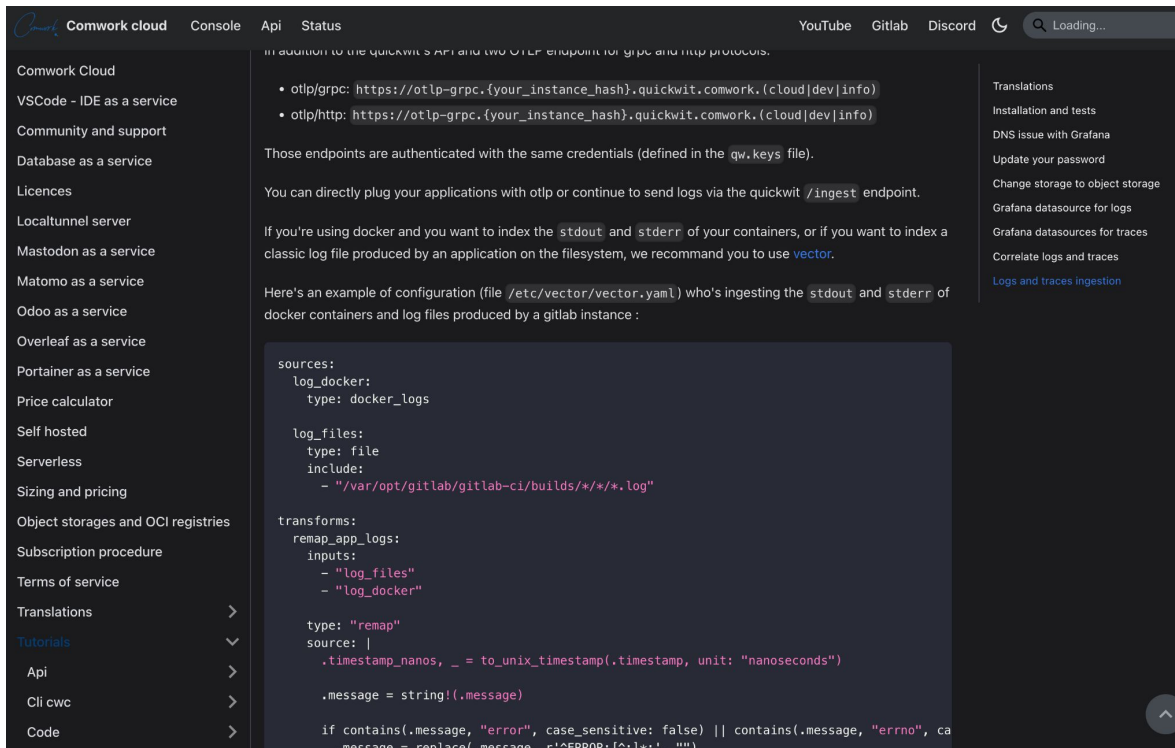


Site web : vector.dev



Comment utiliser Vector avec Quickwit ?

Tutoriel pour rendre les logs avec la définition de l'indexe otel-logs par défaut



Comwork Cloud Console

API Status

YouTube Gitlab Discord

Loading...

Comwork Cloud

VSCoDe - IDE as a service

Community and support

Database as a service

Licences

Localtunnel server

Mastodon as a service

Matomo as a service

Odoo as a service

Overleaf as a service

Portainer as a service

Price calculator

Self hosted

Serverless

Sizing and pricing

Object storages and OCI registries

Subscription procedure

Terms of service

Translations

Tutorials

Api

Cli cwc

Code

In addition to the Quickwit's API and two OTLP endpoints for gRPC and HTTP protocols:

- otlp/gRPC: `https://otlp-grpc.{your_instance_hash}.quickwit.comwork.{cloud|dev|info}`
- otlp/http: `https://otlp-grpc.{your_instance_hash}.quickwit.comwork.{cloud|dev|info}`

Those endpoints are authenticated with the same credentials (defined in the `qw.keys` file).

You can directly plug your applications with OTLP or continue to send logs via the Quickwit `/ingest` endpoint.

If you're using Docker and you want to index the `stdout` and `stderr` of your containers, or if you want to index a classic log file produced by an application on the filesystem, we recommend you to use **vector**.

Here's an example of configuration (file `/etc/vector/vector.yaml`) who's ingesting the `stdout` and `stderr` of Docker containers and log files produced by a GitLab instance :

```
sources:
  log_docker:
    type: docker_logs

  log_files:
    type: file
    include:
      - "/var/opt/gitlab/gitlab-ci/builds/*/*/*.log"

transforms:
  remap_app_logs:
    inputs:
      - "log_files"
      - "log_docker"

    type: "remap"
    source: |
      .timestamp_nanos, _ = to_unix_timestamp(timestamp, unit: "nanoseconds")

      .message = string!(.message)

      if contains(.message, "error", case_sensitive: false) || contains(.message, "errno", case_sensitive: false) {
        .message = replace(.message, r'^(ERROR|ERRNO):', '')
      }
```

Translations

Installation and tests

DNS issue with Grafana

Update your password

Change storage to object storage

Grafana datasource for logs

Grafana datasources for traces

Correlate logs and traces

Logs and traces ingestion

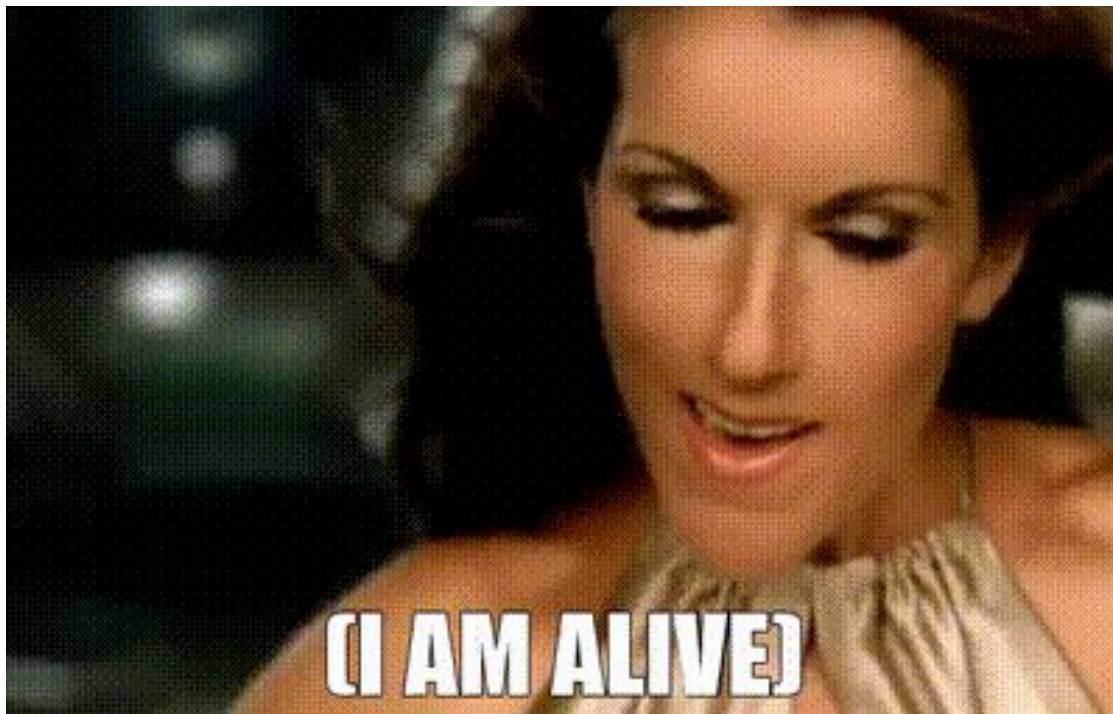
Tutoriel :

doc.cloud.comwork.io/docs/tutorials/quickwit#logs-and-traces-ingestion



Qu'est-ce que imalive ?

Microservice qui exporte les métriques d'une machines (RAM, CPU, Disk)
Compatible Prometheus, OpenTelemetry et écrit également des logs sur stdout
Produit un heartbeat également ainsi qu'une liste de healthcheck configurables



Repo :

gitlab.comwork.io/oss/imalive



Démo

Et si on passait aux choses sérieuses ?

QUICKWIT quickwit-default-cluster Docs

Discover
Query editor
Admin
Indexes
Cluster
Node info
API

Index ID
otel-traces-v0_7

Fields

- trace_id
- trace_state
- service_name
- resource_attributes
- resource_dropped_attributes_count
- scope_name
- scope_version
- scope_attributes
- scope_dropped_attributes_count
- span_id
- span_kind
- span_name
- span_fingerprint
- span_start_timestamp_nanos
- span_end_timestamp_nanos
- span_duration_millis
- span_attributes

RUN

1

13 hits found in 0.01 seconds

```
> 2024/09/13 12:49:27 {
  "resource_attributes": {
    "telemetry.sdk.language": "python",
    "telemetry.sdk.name": "opentelemetry",
    "telemetry.sdk.version": "1.27.0"
  },
  "scope_name": "ut
ils.otel",
  "service_name": "imalive-grafana-imalive",
  "span_duration_millis": 0,
  "span_end_timestamp_nanos": 1726231767360967000,
  "span_fingerprint": "imaliv
e-grafana-imalive",
  "span_id": "b46321d8f2dd395",
  "span_kind": 1,
  "span_name": "imalive-monitors",
  "span_start_timestamp_nanos": 17262317673
60749000,
  "trace_id": "81fbcf36439d3d3e5992aa29287f1781"
}

> 2024/09/13 12:49:17 {
  "resource_attributes": {
    "telemetry.sdk.language": "python",
    "telemetry.sdk.name": "opentelemetry",
    "telemetry.sdk.version": "1.27.0"
  },
  "scope_name": "ut
ils.otel",
  "service_name": "imalive-grafana-imalive",
  "span_duration_millis": 0,
  "span_end_timestamp_nanos": 1726231757359066000,
  "span_fingerprint": "imaliv
e-grafana-imalive",
  "span_id": "5c260beccf43853e",
  "span_kind": 1,
  "span_name": "imalive-monitors",
  "span_start_timestamp_nanos": 17262317573
58842000,
  "trace_id": "6b7b1853261adf860a32af423a769b80"
}

> 2024/09/13 12:49:09 {
  "resource_attributes": {
    "telemetry.sdk.language": "python",
    "telemetry.sdk.name": "opentelemetry",
    "telemetry.sdk.version": "1.27.0"
  },
  "scope_name": "ut
ils.otel",
  "service_name": "imalive-grafana-imalive",
  "span_duration_millis": 0,
  "span_end_timestamp_nanos": 1726231749134299000,
  "span_fingerprint": "imaliv
e-grafana-imalive",
  "span_id": "01c3689c0339860e",
  "span_kind": 1,
  "span_name": "imalive-monitors",
  "span_start_timestamp_nanos": 17262317491
34210000,
  "trace_id": "0d28b11a648607fd70111228f81402cd"
}

> 2024/09/13 12:48:59 {
  "resource_attributes": {
    "telemetry.sdk.language": "python",
    "telemetry.sdk.name": "opentelemetry",
    "telemetry.sdk.version": "1.27.0"
  },
  "scope_name": "ut
ils.otel",
  "service_name": "imalive-grafana-imalive",
  "span_duration_millis": 0,
  "span_end_timestamp_nanos": 1726231739133437000,
  "span_fingerprint": "imaliv
e-grafana-imalive",
  "span_id": "63d19a6d1db9c536",
  "span_kind": 1,
  "span_name": "imalive-monitors",
  "span_start_timestamp_nanos": 17262317391
33196000,
  "trace_id": "c218f0db67641f69b6c561f58b8b331"
}

> 2024/09/13 12:48:59 {
  "resource_attributes": {
    "telemetry.sdk.language": "python",
    "telemetry.sdk.name": "opentelemetry",
    "telemetry.sdk.version": "1.27.0"
  },
  "scope_name": "ut
ils.otel",
  "service_name": "imalive-grafana-imalive",
  "span_duration_millis": 12026,
  "span_end_timestamp_nanos": 1726231751149173000,
  "span_fingerprint": "imaliv
e-grafana-imalive",
  "span_id": "6aafa72599e44088",
  "span_kind": 1,
  "span_name": "imalive-heartbit",
  "span_start_timestamp_nanos": 1726231
739122791000,
  "trace_id": "c14a04ea75ce818f7ae949e627a80665"
}

> 2024/09/13 12:48:52 {
  "resource_attributes": {
    "telemetry.sdk.language": "python",
    "telemetry.sdk.name": "opentelemetry",
    "telemetry.sdk.version": "1.27.0"
  },
  "scope_name": "ut
ils.otel",
  "service_name": "imalive-grafana-imalive",
  "span_duration_millis": 0,
  "span_end_timestamp_nanos": 1726231732709895000,
  "span_fingerprint": "imaliv
e-grafana-imalive",
  "span_id": "d7e4dc5a0740055c",
  "span_kind": 1,
  "span_name": "imalive-monitors",
  "span_start_timestamp_nanos": 17262317327
09803000,
  "trace_id": "d0d133bbe08a19464e33d539f559b8b"
}
```

Repo :

gitlab.comwork.io/comwork_public/talks/bdx-quickwit



Comwork

Merci !

