



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

Voxxed Days Luxembourg, 21/06/2025

Who am I ?

Idriss Neumann

Founder and CTO of cwcloud.tech

SRE/Platform Engineer

Contributeur OSS



idrissneumann

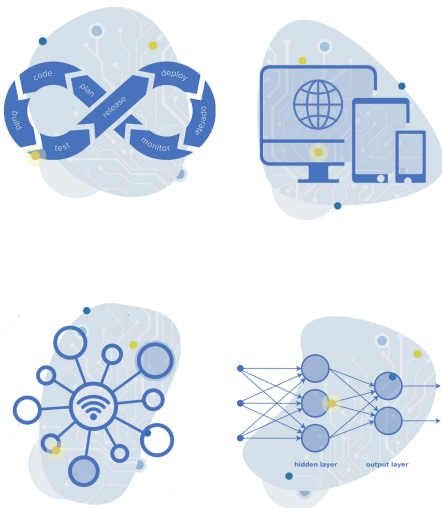
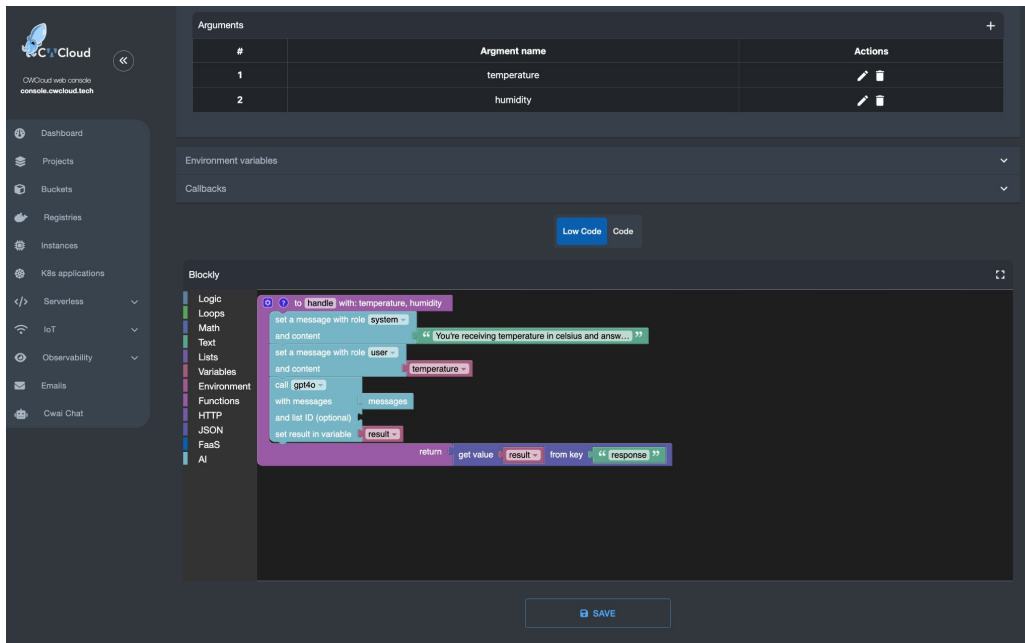


ineumann.fr





Qui sommes nous ?

Editeur logiciel basé à Paris et Tunis

Plateforme DaaS multicloud, FaaS et ML/ops pour accélérer vos déploiements et vos développements

The screenshot displays the CWCloud console interface. On the left is a sidebar menu with options: Dashboard, Projects, Buckets, Registries, Instances, K8s applications, Serverless, IoT, Observability, Emails, and CWat Chat. The main area shows the configuration for a function named 'to handle with: temperature, humidity'. It includes a table for arguments, environment variables, and a visual logic builder (Blockly) for the function's execution flow.

#	Argument name	Actions
1	temperature	 
2	humidity	 

Environment variables and Callbacks sections are also visible but currently empty.

The logic builder shows a sequence of steps: 'to handle with: temperature, humidity', 'set a message with role system', 'and content', 'set a message with role user', 'and content', 'call gpt4o', 'with messages', 'and list ID (optional)', 'set result in variable', 'return', 'get value', 'result = from key', and 'response'.

Website: cwcloud.tech



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

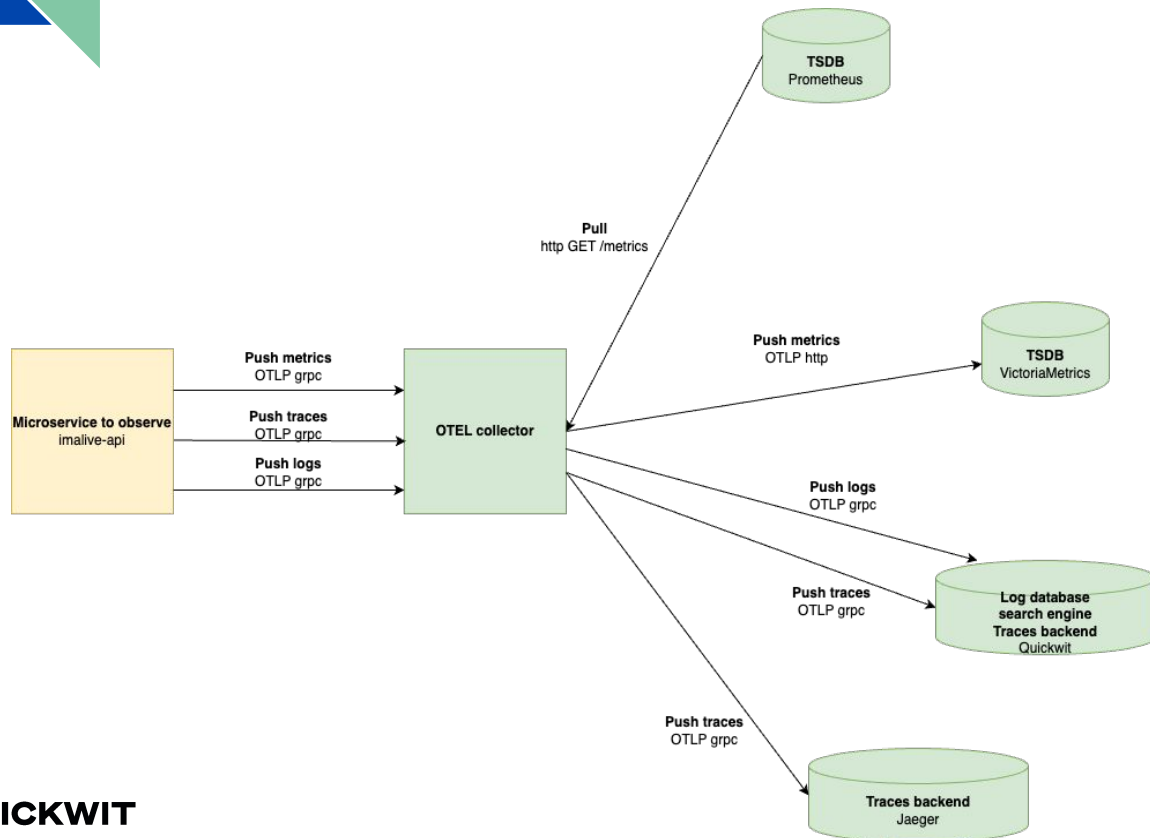
Observability landscape

Classement des outils d'observabilité les plus célèbres



Qu'est-ce qu'OpenTelemetry ?

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

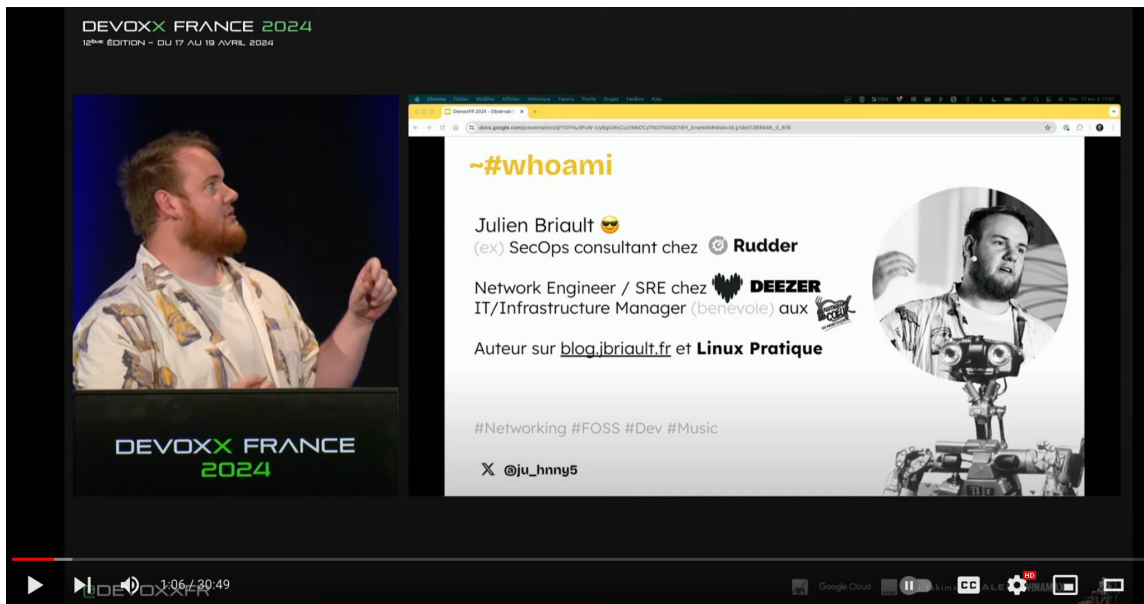


Website: opentelemetry.io



Qu'est-ce que VictoriaMetrics ?

Petite parenthèse pour aller voir le talk de Julien

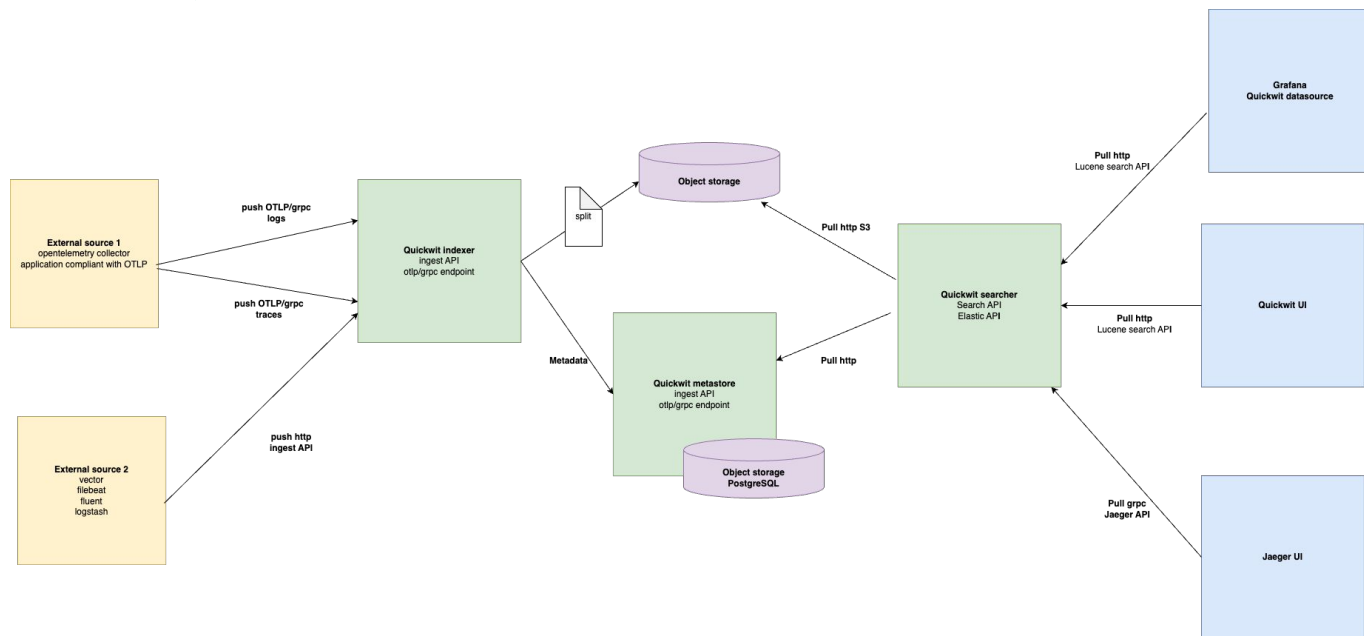


Talk de Julien "Observabilité :
dépoussiérer Prometheus
avec VictoriaMetrics":
youtu.be/bzLfwjUj2k0



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



Website: quickwit.io



Pourquoi choisir Quickwit ?

Les raisons de notre choix de cette solution



[Blog](#) [Documentation](#) [Sign in](#) [English](#)

Recent posts

2025

New identity for CWCloud

DevOps is dead, is it serious doctor?

2024

Replace Google Analytics with Grafana, Quickwit and CWCloud

Installing CWCloud on K8S is so easy!

Quickwit for prometheus metrics

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
founder cwcloud.tech



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.

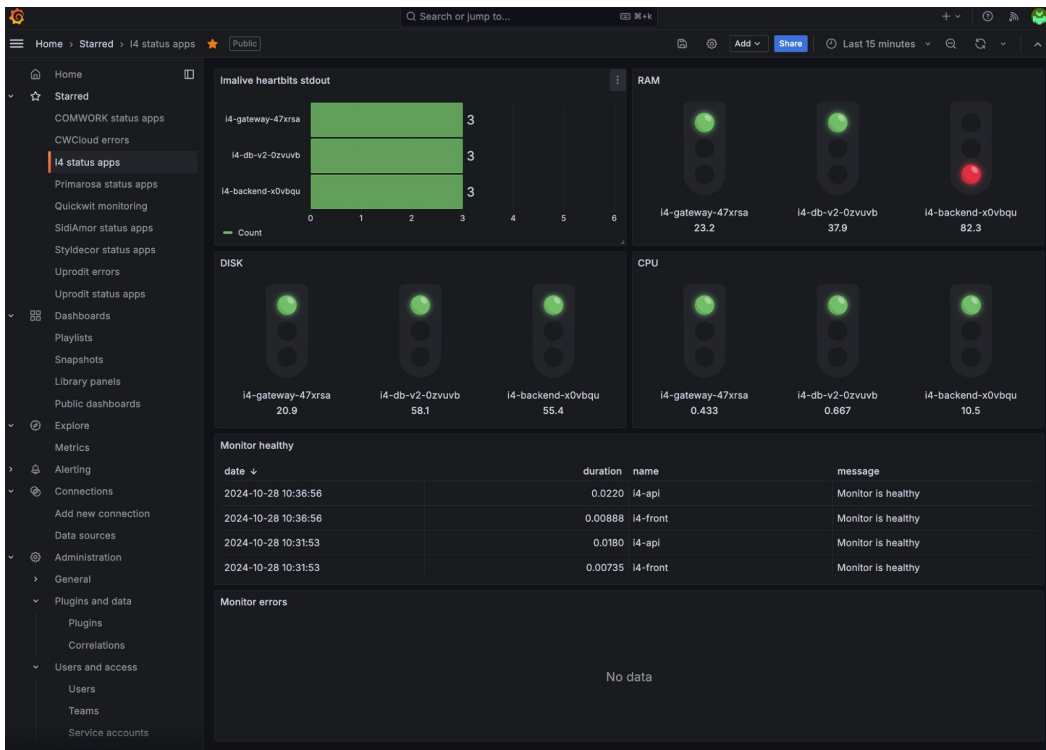


Link: cwcloud.tech/blog/quickwit



Quickwit pour les métriques prometheus ?

Nous avons également fait ce choix et expliquons les avantages et inconvénients

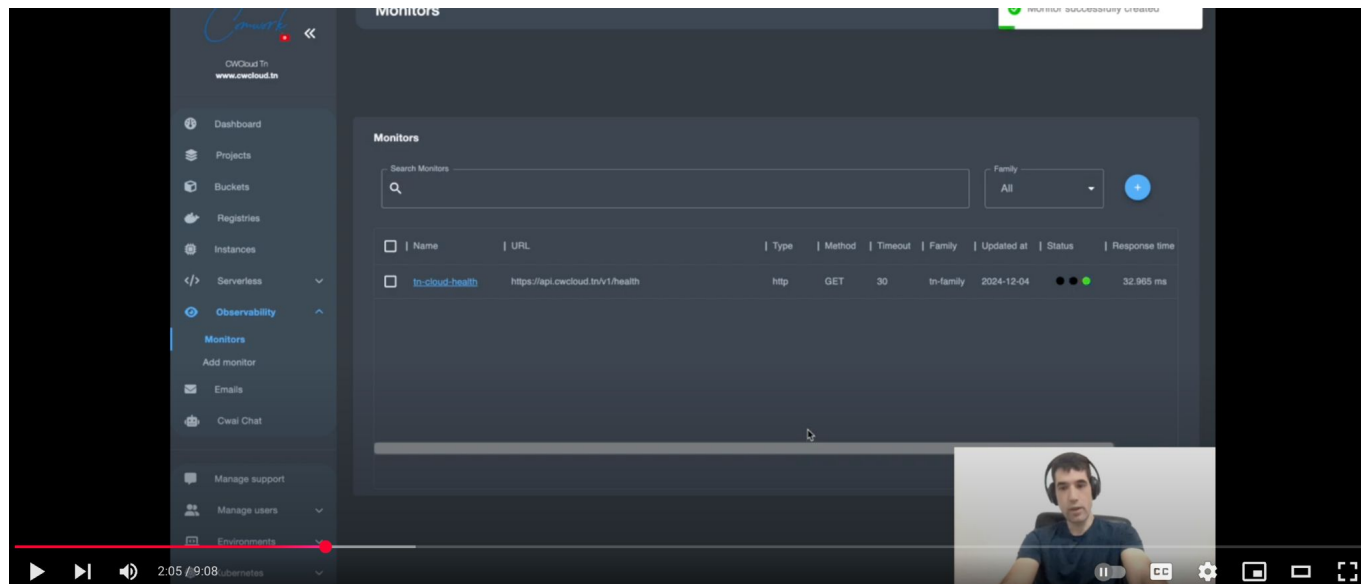


Link: cwcloud.tech/blog/quickwit-metrics



Quickwit pour les métriques prometheus ?

Démo en utilisant les fonctionnalités de monitoring de CWCloud



The screenshot shows the CWCloud Monitors interface. On the left is a sidebar with navigation options: Dashboard, Projects, Buckets, Registries, Instances, Serverless, Observability (selected), Monitors, Add monitor, Emails, Owl Chat, Manage support, Manage users, and Environments. The main panel is titled 'Monitors' and contains a search bar, a 'Family' dropdown set to 'All', and a table of monitors.

<input type="checkbox"/>	Name	URL	Type	Method	Timeout	Family	Updated at	Status	Response time
<input type="checkbox"/>	tn-cloud-health	https://api.cwcloud.tn/v1/health	http	GET	30	tn-family	2024-12-04	●	32.965 ms

At the bottom right of the interface is a video feed of a person wearing headphones, likely the presenter. The video player controls at the bottom show a progress bar at 2:05 / 9:08.

English version: youtu.be/dpqbhpzVXmo



French version: youtu.be/DYu6m1JQ-ds



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

Composite types

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

Link :

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`

Link:

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)
```

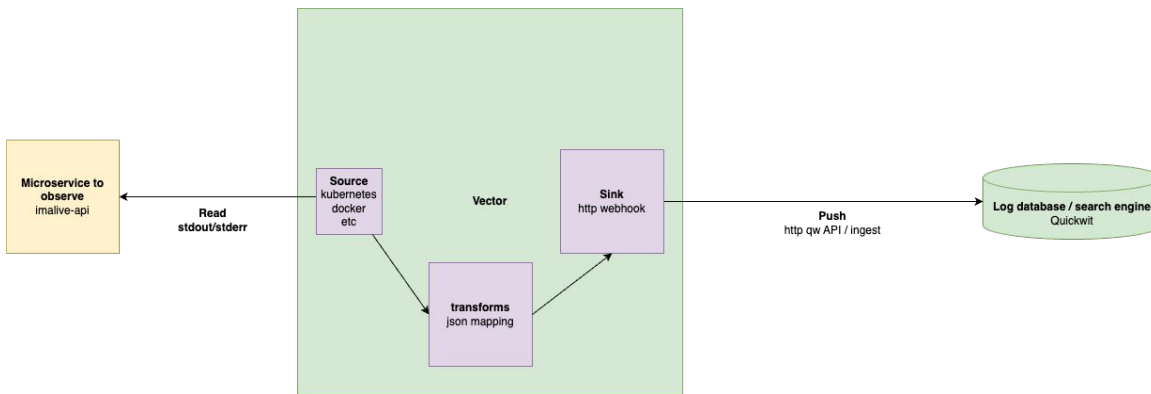
Link:

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

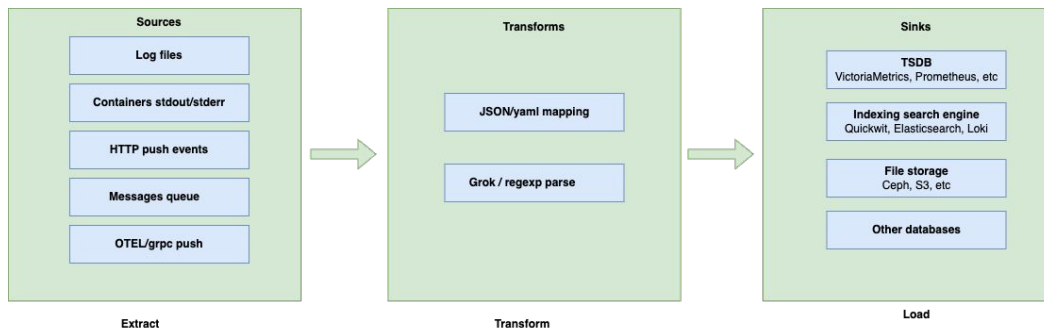


Qu'est-ce que Vector ?

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

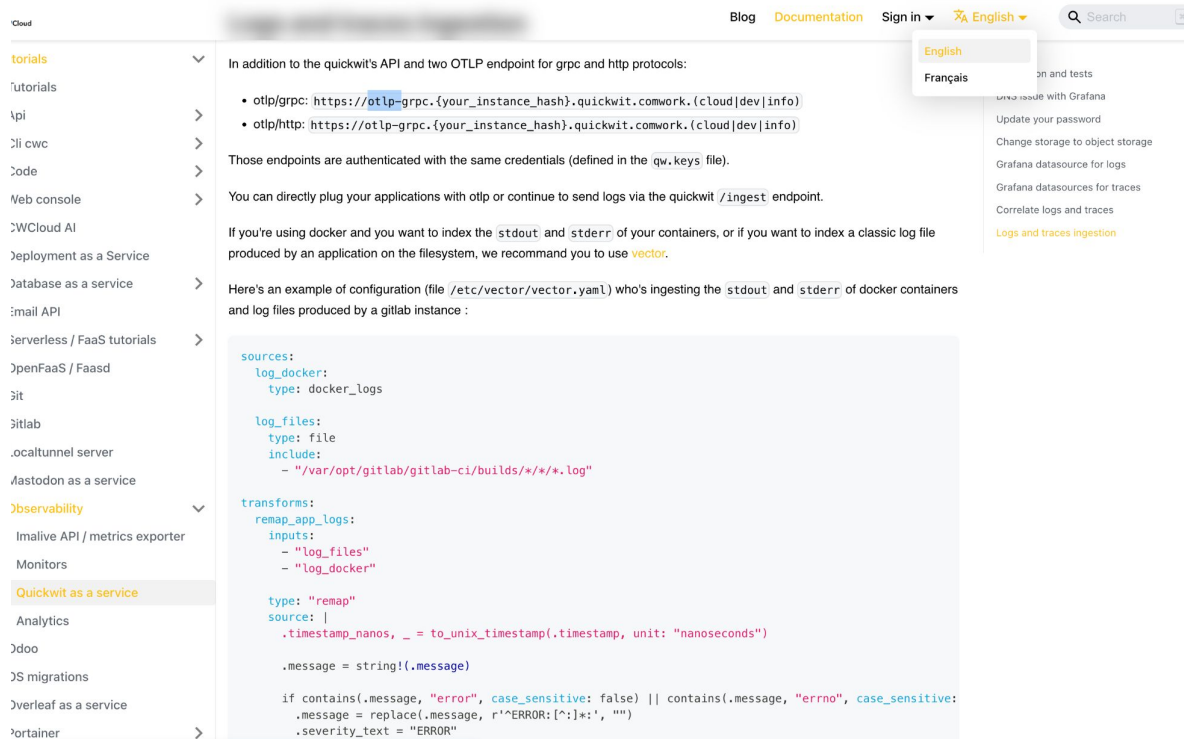


Website: vector.dev



Comment utiliser Vector avec Quickwit ?

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



Cloud

Blog Documentation Sign in English Français

Search

tutorials

Tutorials

API

CLI cwc

Code

Web console

Cloud AI

Deployment as a Service

Database as a service

Mail API

Serverless / FaaS tutorials

OpenFaaS / Faasd

Git

Gitlab

Local tunnel server

Fastodot as a service

Observability

Malware API / metrics exporter

Monitors

Quickwit as a service

Analytics

Dojo

JS migrations

Verleaf as a service

Container

In addition to the quickwit's API and two OTLP endpoint for grpc and http protocols:

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

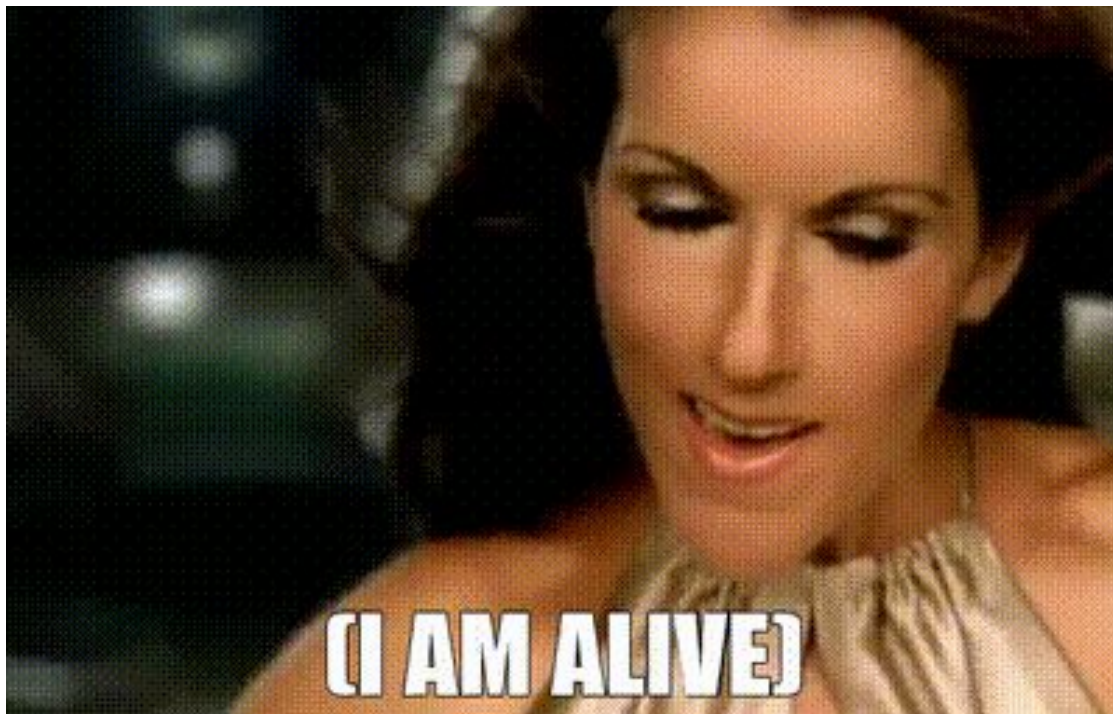
      if contains(.message, "error", case_sensitive: false) || contains(.message, "errno", case_sensitive:
        .message = replace(.message, r'^ERROR: [^:]*:', '')
        .severity_text = "ERROR"
```

Tutorial:
cwcloud.tech/docs/tutorials/observability/quickwit



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



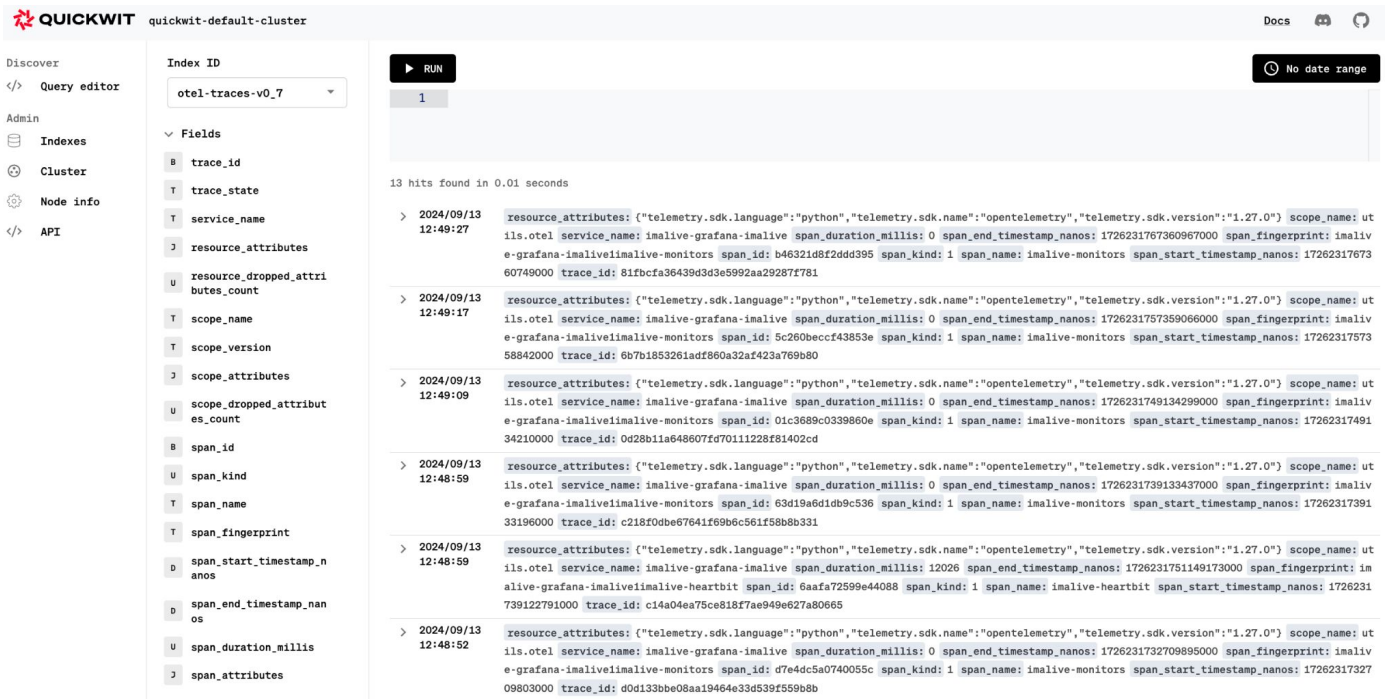
Repo :

gitlab.comwork.io/oss/imalive



Démo

Et si on passait aux choses sérieuses ?


quickwit-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

No date range

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: utils.otel service_name: imalive-grafana-imalive span_duration_millis: 0 span_end_timestamp_nanos: 1726231767360967000 span_fingerprint: imalive-grafana-imalive-imalive-monitors span_id: b46321d8f2dd395 span_kind: 1 span_name: imalive-monitors span_start_timestamp_nanos: 1726231767360749000 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: utils.otel service_name: imalive-grafana-imalive span_duration_millis: 0 span_end_timestamp_nanos: 1726231757359066000 span_fingerprint: imalive-grafana-imalive-imalive-monitors span_id: 5c260beccf43853e span_kind: 1 span_name: imalive-monitors span_start_timestamp_nanos: 1726231757358842000 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: utils.otel service_name: imalive-grafana-imalive span_duration_millis: 0 span_end_timestamp_nanos: 1726231749134299000 span_fingerprint: imalive-grafana-imalive-imalive-monitors span_id: 01c3689c0339860e span_kind: 1 span_name: imalive-monitors span_start_timestamp_nanos: 1726231749134210000 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: utils.otel service_name: imalive-grafana-imalive span_duration_millis: 0 span_end_timestamp_nanos: 1726231739133437000 span_fingerprint: imalive-grafana-imalive-imalive-monitors span_id: 63d19a6d1db9c536 span_kind: 1 span_name: imalive-monitors span_start_timestamp_nanos: 1726231739133196000 trace_id: c218f0dbe67641f69b6c561f58b8b331

2024/09/13 12:48:59

resource_attributes: {"telemetry.sdk.language": "python", "telemetry.sdk.name": "opentelemetry", "telemetry.sdk.version": "1.27.0"} scope_name: utils.otel service_name: imalive-grafana-imalive span_duration_millis: 12026 span_end_timestamp_nanos: 1726231751149173000 span_fingerprint: imalive-grafana-imalive-imalive-heartbit span_id: 6aafa72599e44088 span_kind: 1 span_name: imalive-heartbit span_start_timestamp_nanos: 1726231739122791000 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: utils.otel service_name: imalive-grafana-imalive span_duration_millis: 0 span_end_timestamp_nanos: 1726231732709895000 span_fingerprint: imalive-grafana-imalive-imalive-monitors span_id: d7e4dc5a0740055c span_kind: 1 span_name: imalive-monitors span_start_timestamp_nanos: 1726231732709803000 trace_id: d0d133bbe08aa19464e33d539f559b8b

Repo:

gitlab.com/work.io/comwork_public/talks/voxxedlu-quickwit





Merci !
