FORK IT!

CWCloud

Let's discover together the next generation of observability with logs and traces: Quickwit

Fork-IT in Tunis, 05/04/2025

QUICKWIT

Who am I ?

Idriss Neumann

Founder and CTO of cwcloud.tech

SRE/Platform Engineer specialist

OSS contributor

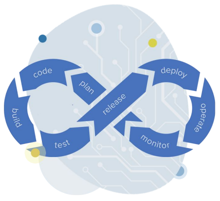
idrissneumann

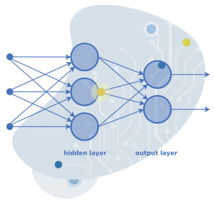
ineumann.fr

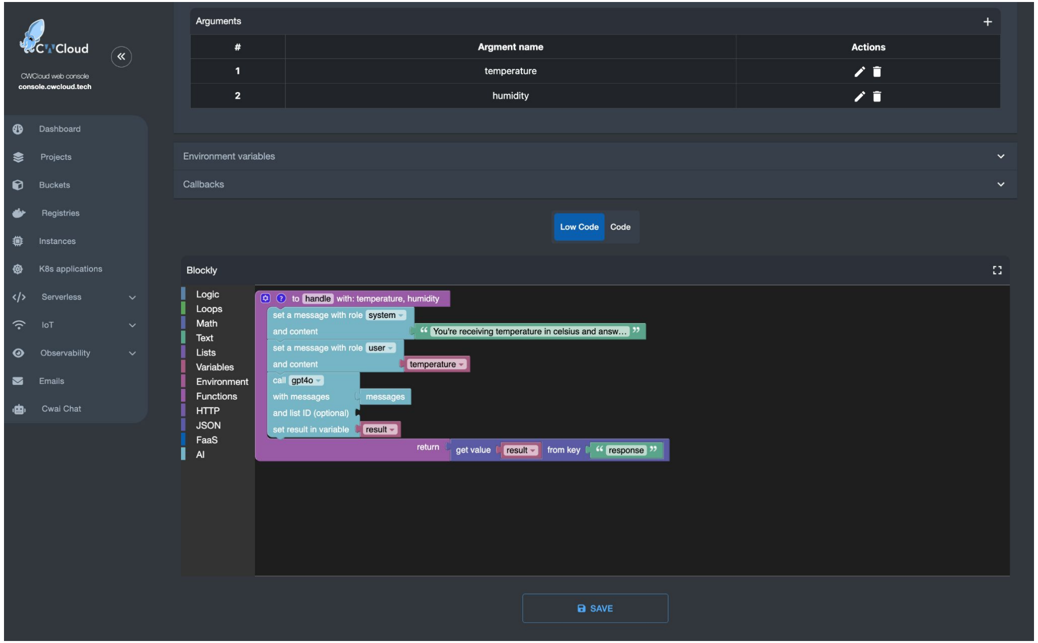
Who are we?

Software editor based in Paris and Tunis

Multicloud DaaS, FaaS and ML/ops platform to accelerate your development and deployment







Arguments

#	Argment name	Actions
1	temperature	
2	humidity	

Blockly

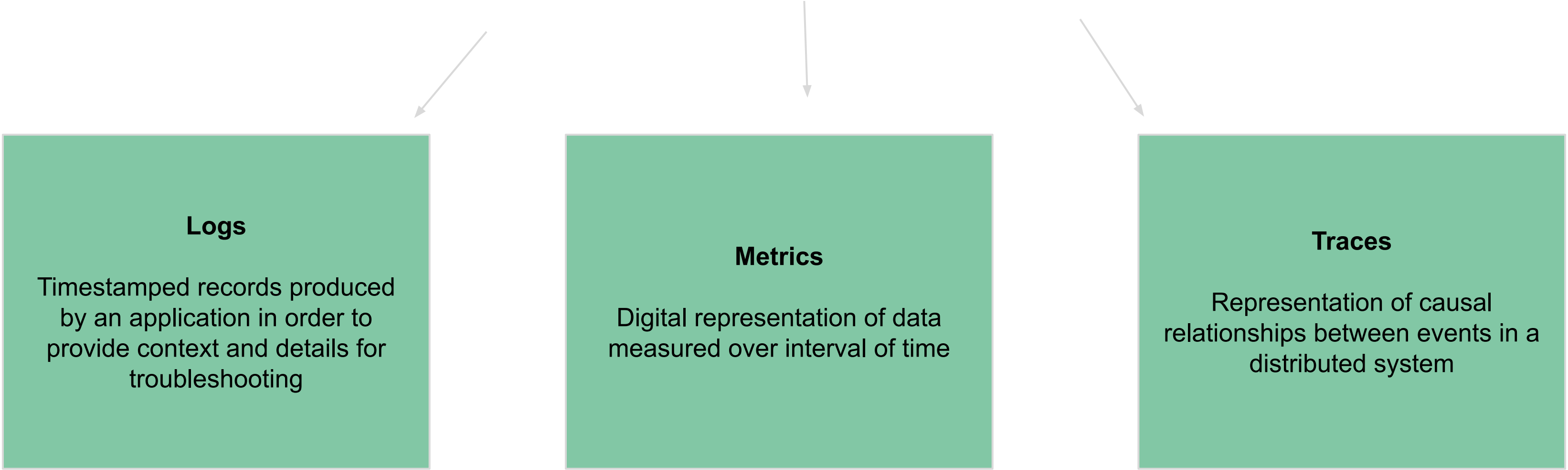
```
to handle with: temperature, humidity
  set a message with role system
  and content "You're receiving temperature in celsius and answ..."
  set a message with role user
  and content temperature
  call gpt4o
  with messages messages
  and list ID (optional)
  set result in variable result
return get value result from key "response"
```

Website: cwcloud.tech

What is observability?

Definition of observability and its three pillars: logs, metrics and traces

Observability is the ability to measure a system's current state based on the data it generates, such as **logs**, **metrics**, and **traces**.



Logs

Timestamped records produced by an application in order to provide context and details for troubleshooting

Metrics

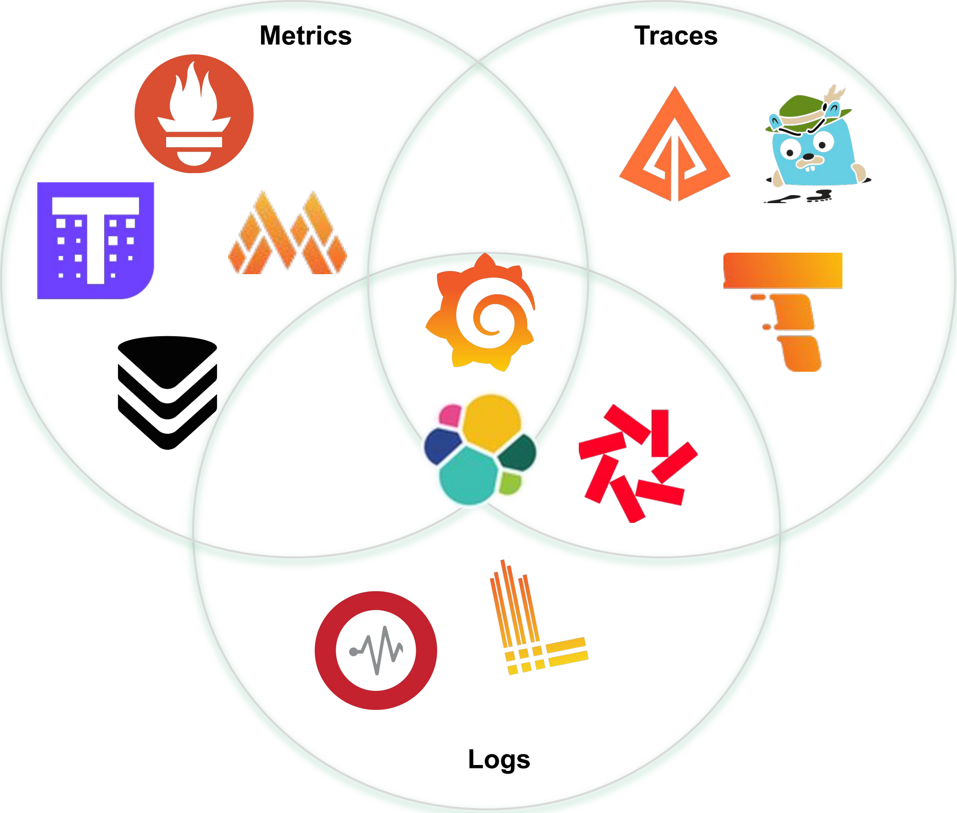
Digital representation of data measured over interval of time

Traces

Representation of causal relationships between events in a distributed system

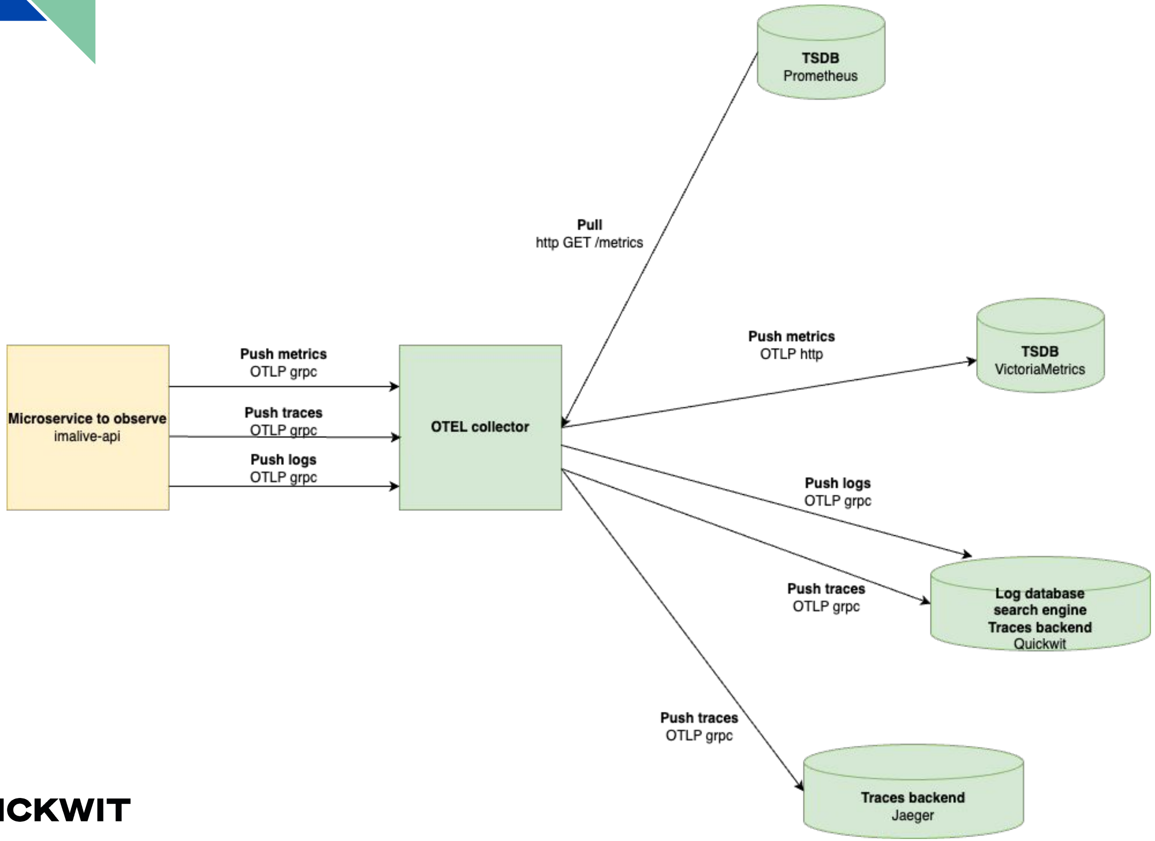
Observability landscape

Most of the well known tools



What is OpenTelemetry?

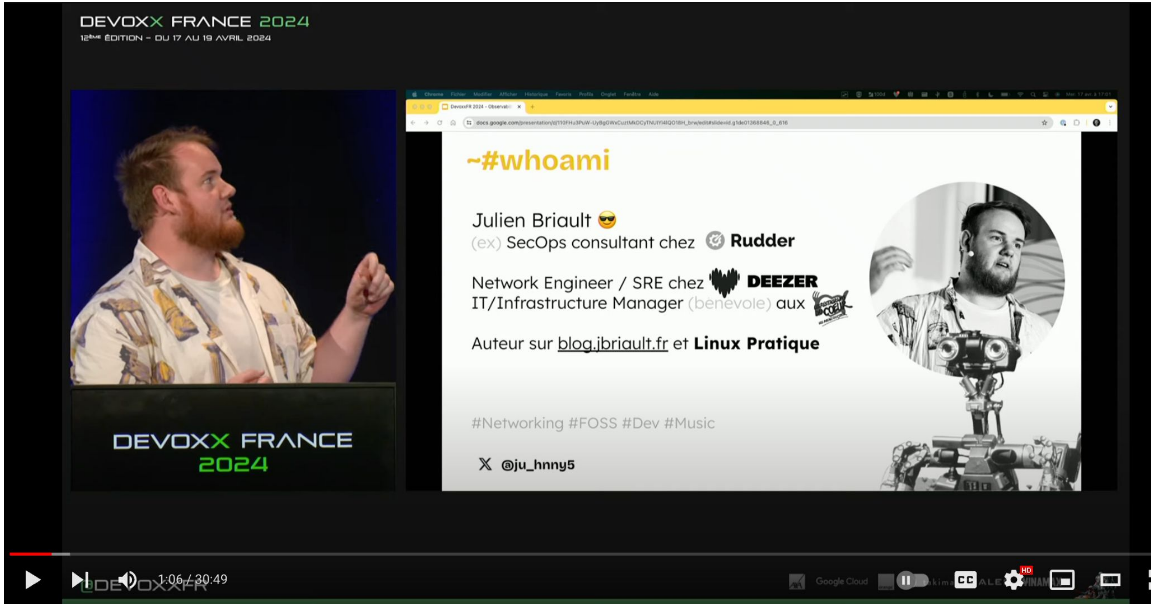
An observability standard for collecting traces, metrics and logs and ensure interoperability



Website: opentelemetry.io

What is VictoriaMetrics?

A quick aside to go see Julien's talk



DEVOXX FRANCE 2024

12ème ÉDITION - DU 17 AU 19 AVRIL 2024

~#whoami

Julien Briault 😎

(ex) SecOps consultant chez Rudder

Network Engineer / SRE chez DEEZER

IT/Infrastructure Manager (bénévole) aux

Auteur sur blog.jbriault.fr et Linux Pratique

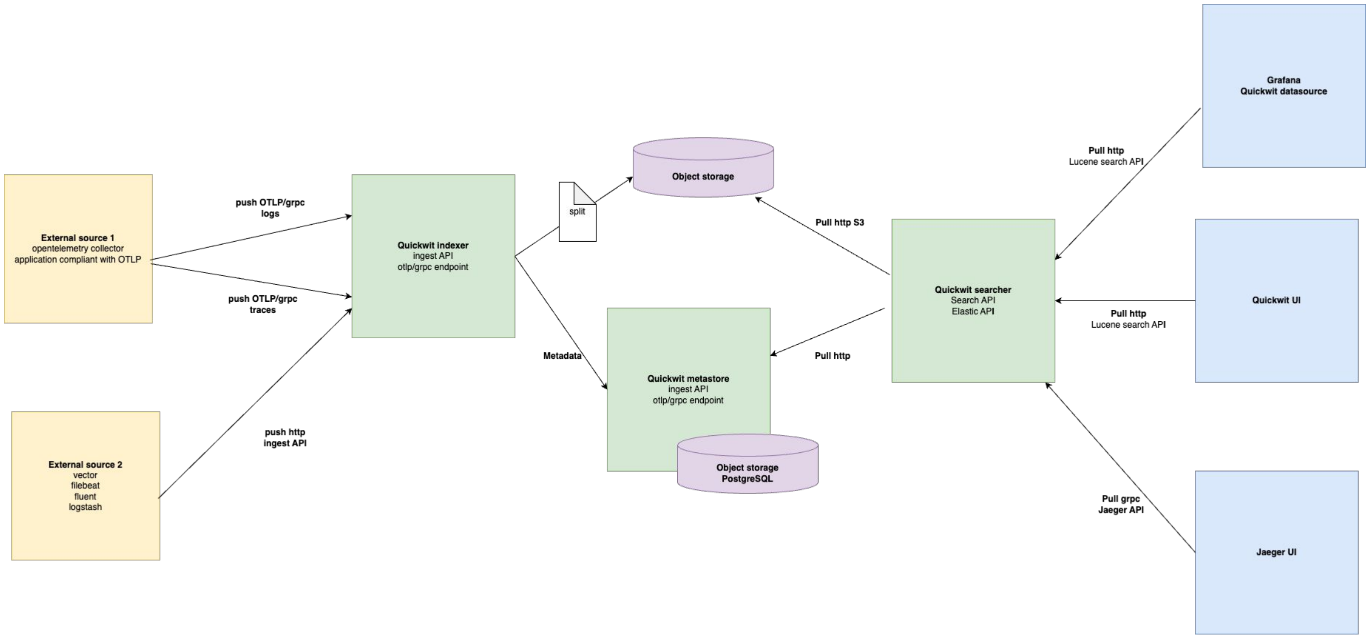
#Networking #FOSS #Dev #Music

X @ju_hnny5

Julien's talk "*Observabilité : dépoussiérer Prometheus avec VictoriaMetrics*": youtu.be/bzLtWjUj2k0

What is Quickwit?

Search engine solution competing with Elasticsearch, OpenSearch, and Grafana Loki
A bit of the best of both worlds combined
Very fast, written in Rust and owned by datadog



External source 1
opentelemetry collector
application compliant with OTLP

External source 2
vector
filebeat
fluent
logstash

push OTLP/grpc
logs

push OTLP/grpc
traces

push http
ingest API

Quickwit indexer
ingest API
otlp/grpc endpoint

split

Object storage

Metadata

Quickwit metastore
ingest API
otlp/grpc endpoint

Object storage
PostgreSQL

Pull http S3

Pull http

Quickwit searcher
Search API
Elastic API

Pull http
Lucene search API

Grafana
Quickwit datasource

Pull http
Lucene search API

Quickwit UI

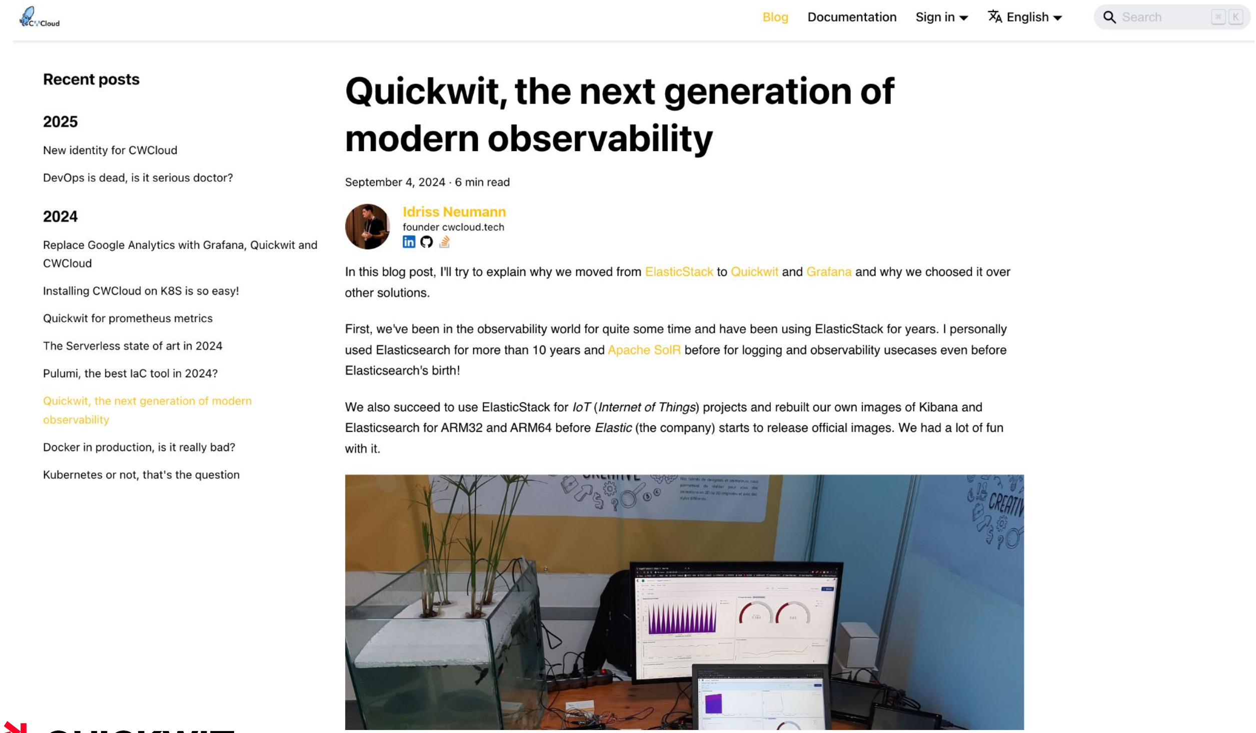
Pull grpc
Jaeger API

Jaeger UI

Website: quickwit.io

Why choosing Quickwit?

The reasons for our choice of this solution



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.

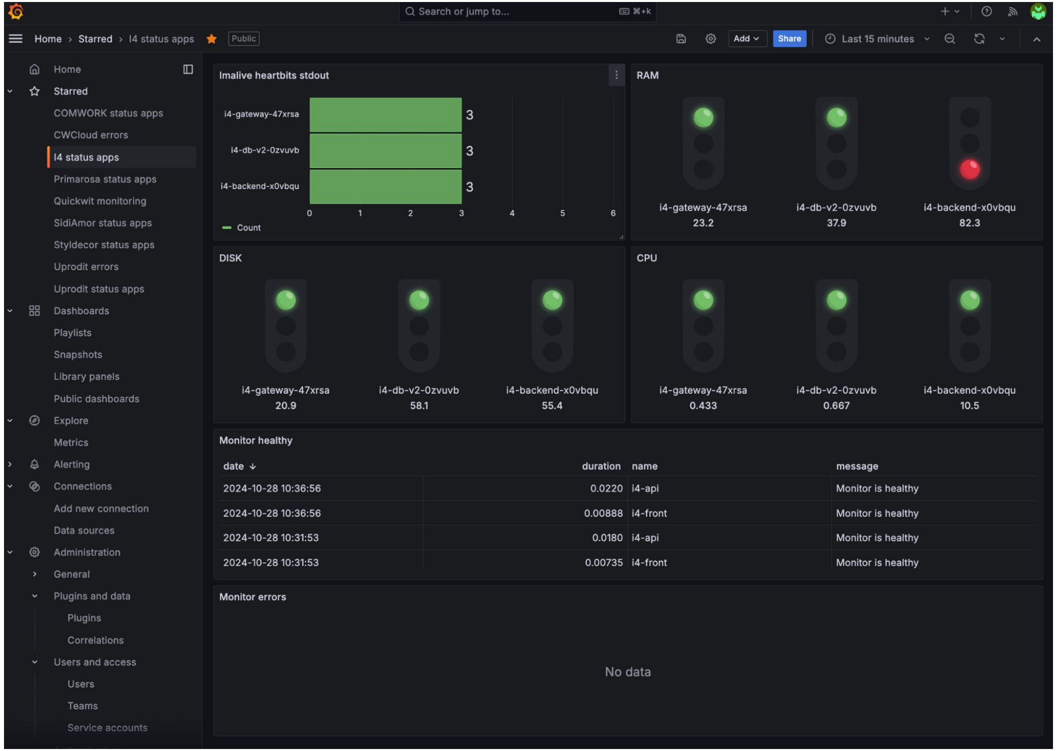
Link: cwcloud.tech/blog/quickwit

QUICKWIT

FORK IT

Quickwit for prometheus metrics?

We have also made this choice and explain the pros and cons



Monitor healthy

date ↓	duration	name	message
2024-10-28 10:36:56	0.0220	i4-api	Monitor is healthy
2024-10-28 10:36:56	0.00888	i4-front	Monitor is healthy
2024-10-28 10:31:53	0.0180	i4-api	Monitor is healthy
2024-10-28 10:31:53	0.00735	i4-front	Monitor is healthy

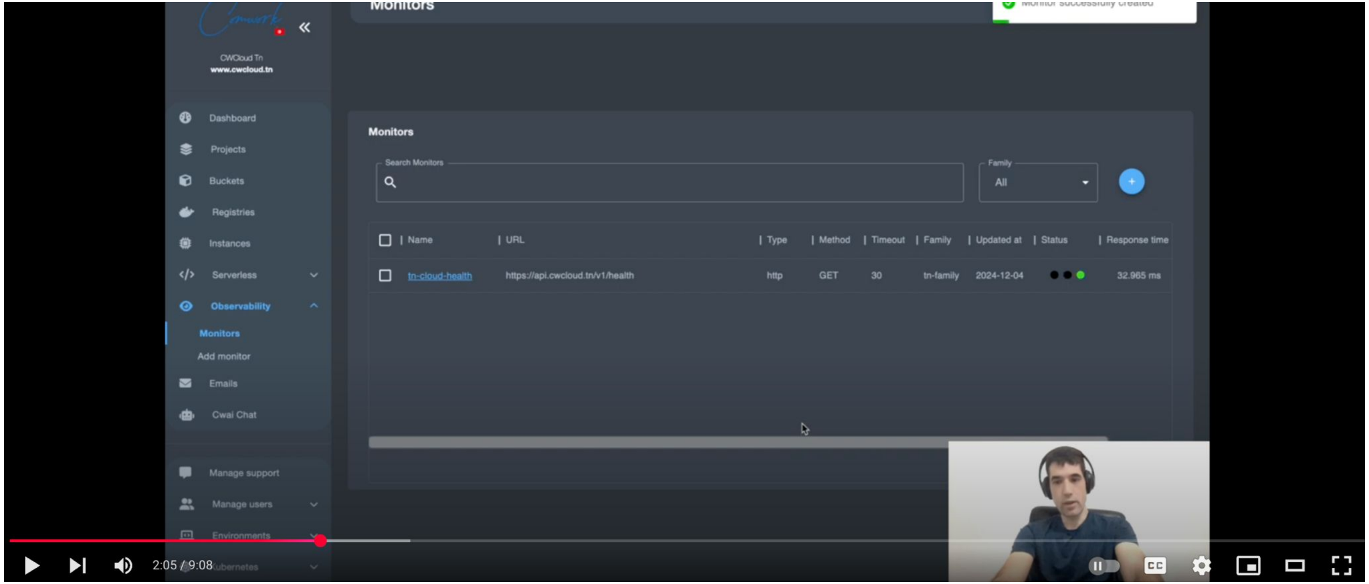
Monitor errors

No data

Link: cwcloud.tech/blog/quickwit-metrics

Quickwit for prometheus metrics?

Demo with the CWCloud's observability features



Monitors

Search Monitors

Family: All

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

English version: youtu.be/dpqbhpzVXmo

French version: youtu.be/DYu6m1JQ-ds

Basics index mappings with Quickwit

Field types

- ➔ `text`: string / plain text
- ➔ `datetime`: date / timestamp
- ➔ `i64`: integer (64 bits)
- ➔ `f64`: floatting number (64 bits)
- ➔ `u64`: unsigned integer (64 bits)
- ➔ `ip`: IP address
- ➔ `bytes`: binary value or base64 representation
- ➔ `json`: dynamic object

Composite types

- ➔ `array`: list of fields
- ➔ `object`: nested object structure

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

Basics Quickwit’s query

Structure of a query

```
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

Basics Quickwit's query

Logical operators

```
NOT field:condition
```

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

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

By default, a AND operator is assumed

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

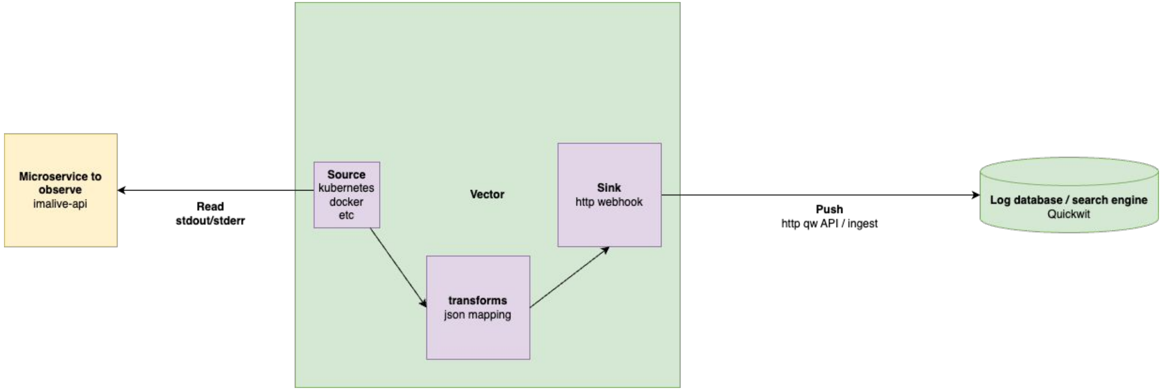
You can also group your queries with parenthesis:

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

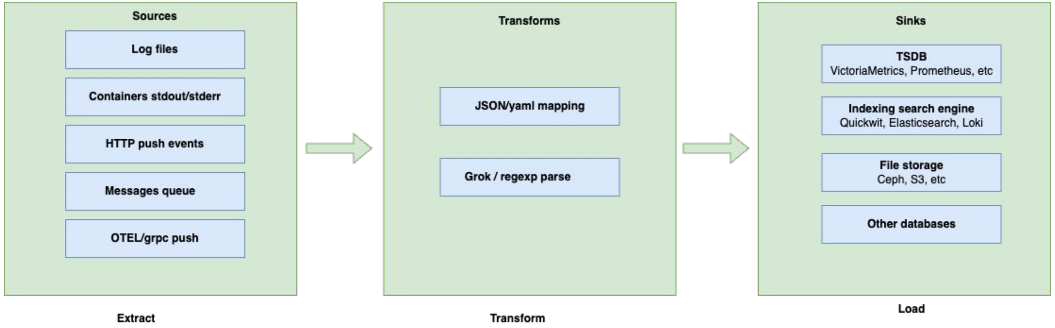
Link:
quickwit.io/docs/get-started/query-language-intro

What is Vector?

Very fast and low footprint observability agent and ETL
Written in Rust and owned by datadog as well

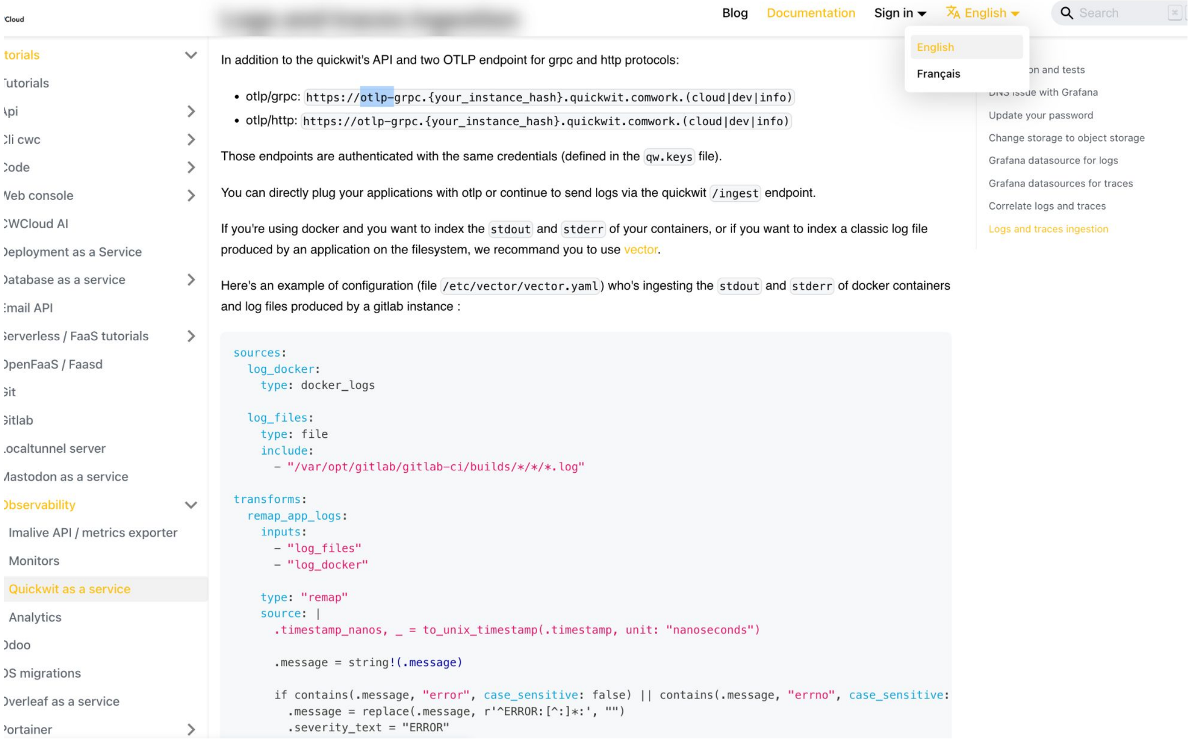


Website: vector.dev



How to use Vector with Quickwit?

Tutorial to collect logs with Vector and index-it in the default otel-logs index



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-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 recommand you to use vector.

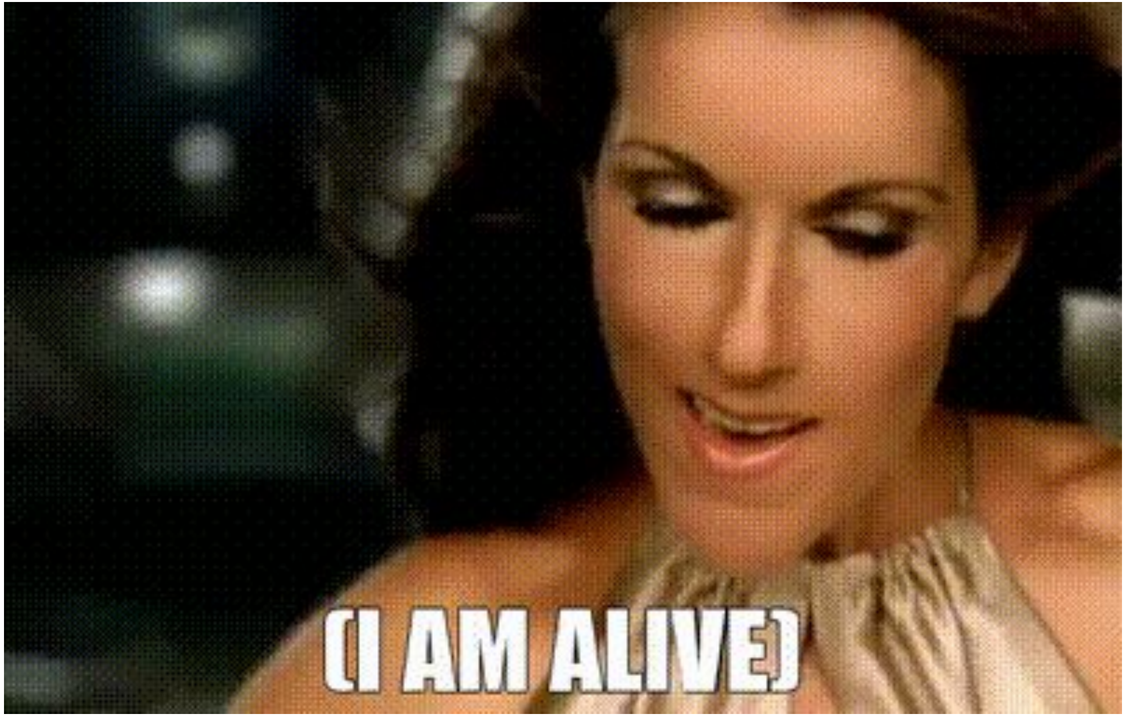
Here's an example of configuration (file `/etc/vector/vector.yaml`) who's ingesting the `stdout` and `stderr` of docker containers and log files produced by a gitlab instance :

```
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

What is Imalive ?

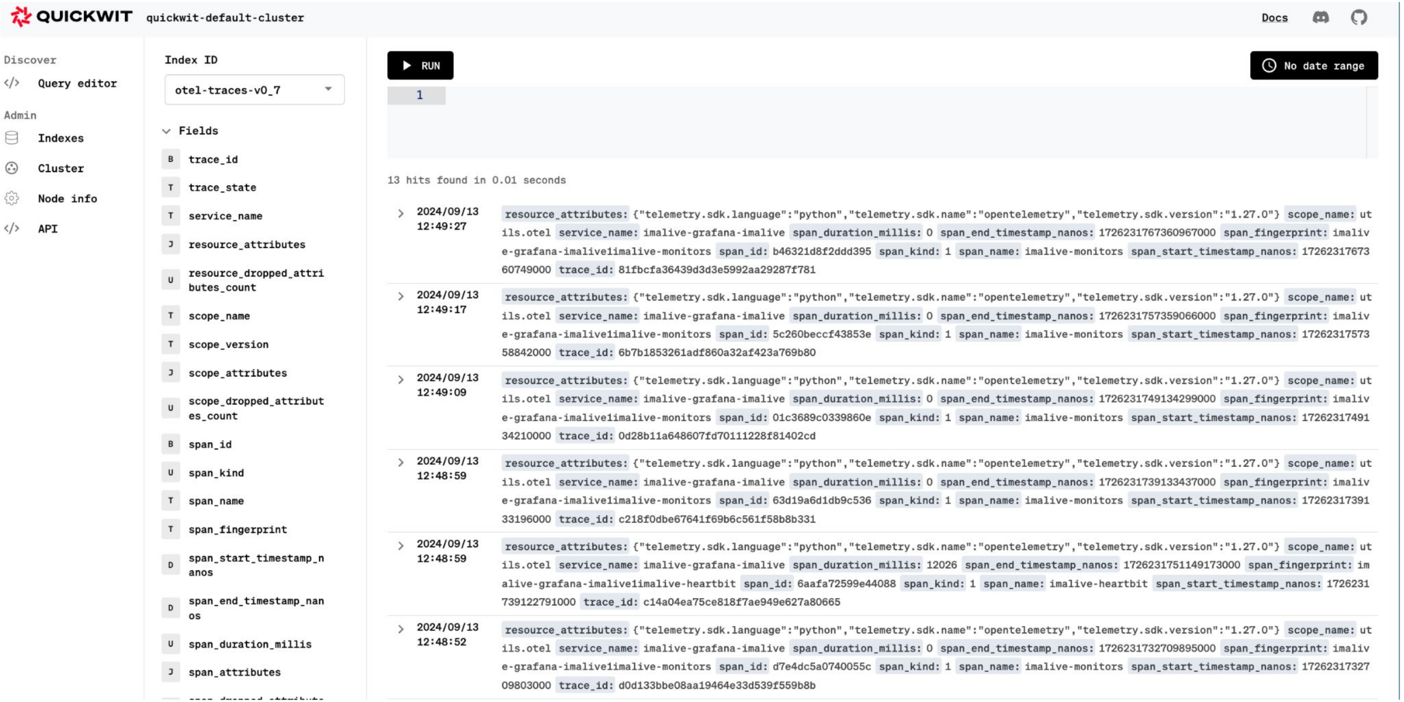
Host metrics exporter (RAM, CPU, Disk) with a heartbit
Compliant with Prometheus / OpenMetrics and OpenTelemetry format



Repo :
gitlab.comwork.io/oss/imalive

Demo

What if we got down to the real deal?



Index ID: otel-traces-v0_7

```
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: utils.otel service_name: imalive-grafana-imalive span_duration_millis: 0 span_end_timestamp_nanos: 1726231767360967000 span_fingerprint: imalive-grafana-imalive1imalive-monitors span_id: b46321d8f2ddd395 span_kind: 1 span_name: imalive-monitors span_start_timestamp_nanos: 1726231767360749000 trace_id: 81fbcfa36439d3d3e5992aa29287f781

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-imalive1imalive-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-imalive1imalive-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-imalive1imalive-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-imalive1imalive-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-imalive1imalive-monitors span_id: d7e4dc5a0740055c span_kind: 1 span_name: imalive-monitors span_start_timestamp_nanos: 1726231732709803000 trace_id: d0d133bbe08aa19464e33d539f559b8b
```

Repo:
gitlab.comwork.io/comwork_public/talks/forkit-quickwit

CWCloud

Thanks !