Observability Day
EUROPE

Unlocking Multi-Cloud Observability

The Case Study of EO4EU Project's Observability Platform

Armagan Karatosun – ECMWF
Francesco Maria Cultrera – CINECA
Lucía Rodríguez Muñoz – CINECA

Who are we?

Observability Day
EUROPE

19 March 2024 | Paris, France

Francesco Maria Cultrera
Cloud Engineer
CINECA

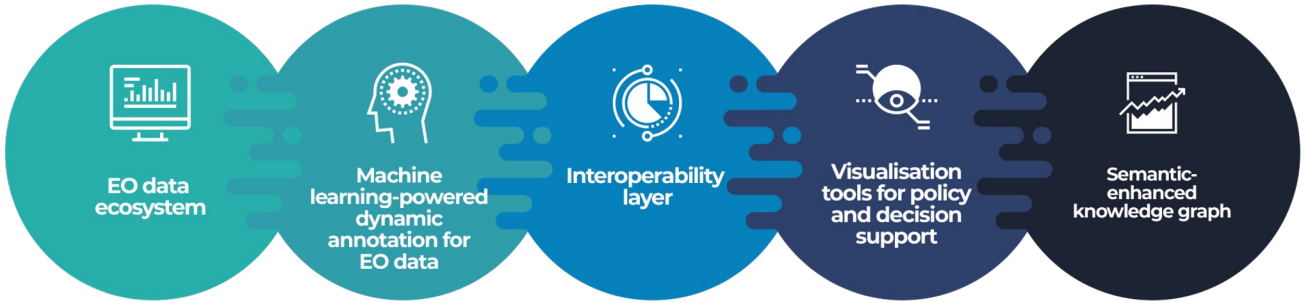
Armagan Karatosun
Cloud Computing Engineer
ECMWF

Introduction: EO4EU Project

Earth Observation for European Union (EO4EU)* is a **European Union-funded** initiative that aims at creating an advanced platform for searching, discovering, processing and analyzing Earth Observation (EO) data and it is currently being developed by a consortium of European partners.

The EO4EU platform is based on a series of innovative technologies which allow to:

- Access** EO data from different sources (e.g., Copernicus, Galileo, ECMWF)
- Support a sophisticated representation of data through a semantic-enhanced knowledge graph
- Use machine learning (ML) from marketplace to EO data processing
- Visualize EO data through easy-to-use graphical interfaces and extended reality (XR) web applications



Funded by the European Union

Work funded by the EU Horizon Europe grant 101060784 (EO4EU)

* https://www.eo4eu.eu/platform

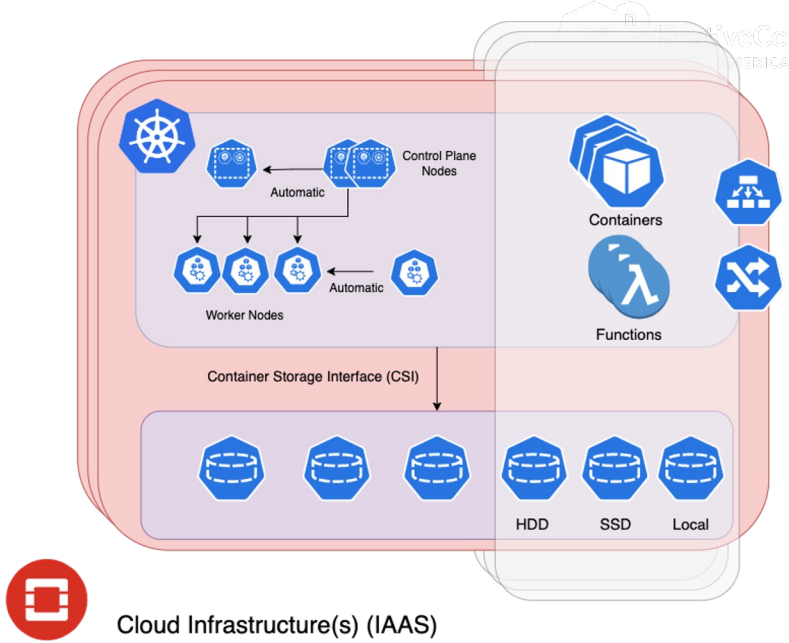
** *Public user access soon-ish (Q1/2024)*

Introduction: EO4EU Architecture

The EO4EU Platform uses a **multi-cloud** and **multi-cluster** architecture that leverages on 2 different OpenStack infrastructures:

- CINECA Supercomputing infrastructure provides High Performance Computing (HPC) and Cloud capabilities with **Leonardo** and **ADA Cloud** systems
- WEkEO*, a part of the Copernicus Data and Information Access Services (DIAS)

offering Infrastructure as a Service (IaaS) functionalities**, and multiple Kubernetes clusters** distributed across them.



ECMWF* *is one of the key partners of the* ***WEkEO* *as Distributed Partner Infrastructure (DPI)*

Pillars of Observability

Observability pillars	Standards
Metrics are numerical values that show how a system is performing over a period of time.	Prometheus with Alertmanager - de-facto standard for metrics in Kubernetes - often referred as **kube-prometheus**.
Logs are immutable and timestamped records of events that happened over time.	Fluentd (with Fluent Bit) and document based database such as OpenSearch/ElasticSearch - often referred as **EFK Stack**.
Traces represent a series of causally related distributed events that encode the end-to-end request flow through a distributed system.	**OpenTelemetry** gained a lot of popularity and adaptation, while still having options to integrate with other **established solutions like Jaeger.**

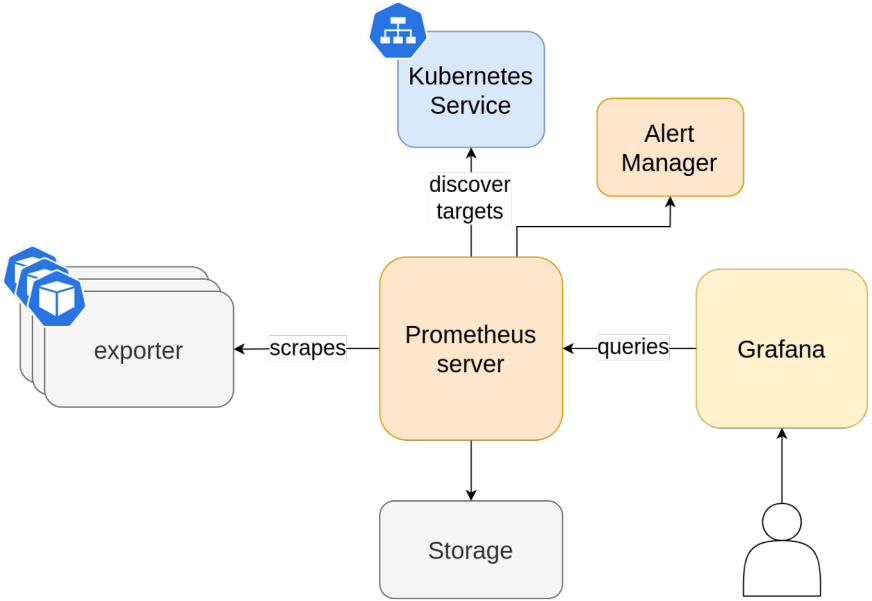
Metrics: Single-cluster Case

Prometheus (kube-prometheus)
Monitoring and alerting system which collects metrics as time series data.

Grafana (grafana-operator)
Popular application to interactively visualize and analyze data.

Challenges in multi-cloud/cluster case:

- Scalability
- Availability
- Historical data
- Centralized alerts



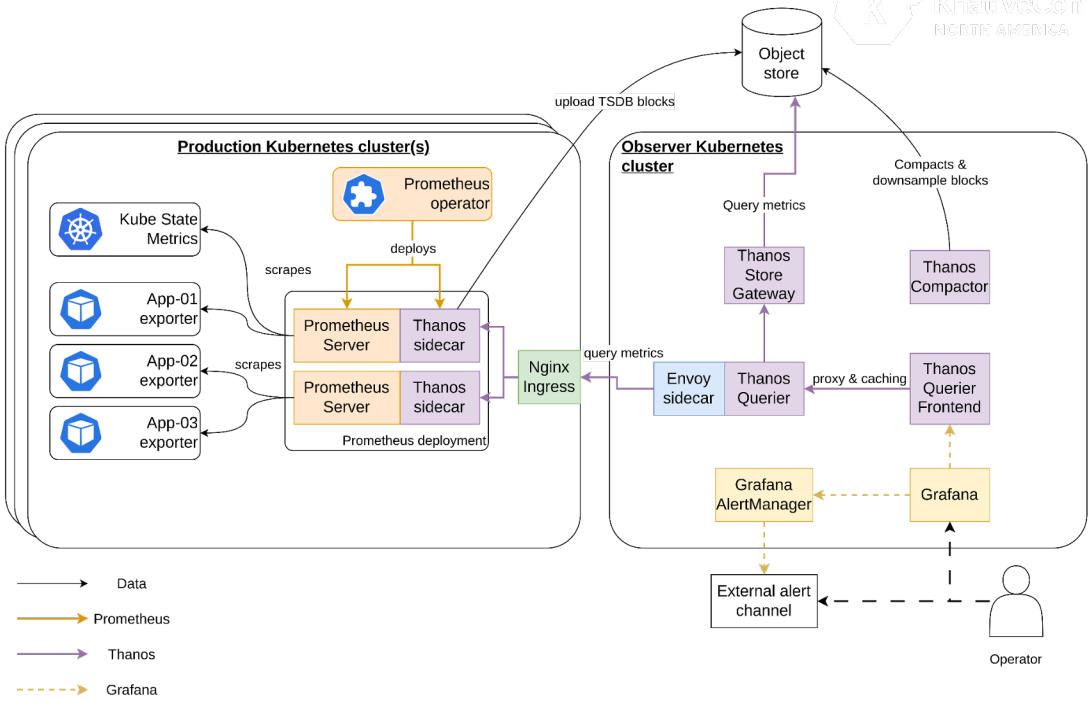
Metrics: Multi-cloud/cluster Case

Thanos (Bitnami Helm chart) provides:

- Prometheus compatibility (Grafana DataSource)
- Global Query View
- Unlimited retention
- Long-term storage compaction
  - Archival data
  - Faster metrics visualization

General issues:

- Sidecar approach management and resource consumption
- Direct query (data stored from 2h)
- Querier requires reverse proxy to reach clusters



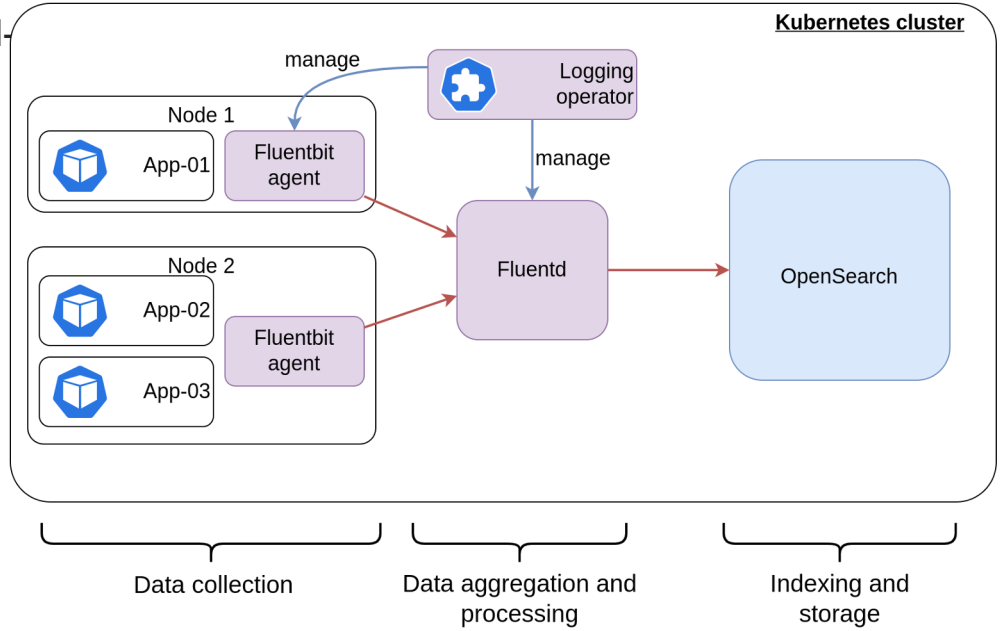
Logs: Single-cluster Case

Fluentd/Fluent Bit (kube-logging/logging-operator)

- De-facto standard for logs (especially on cloud-native)
- Many source and output plugins
- Built-in filters and parsers available for common use-cases (e.g, nginx)
- TLS encryption support

Opensearch (opensearch-k8s-operator)

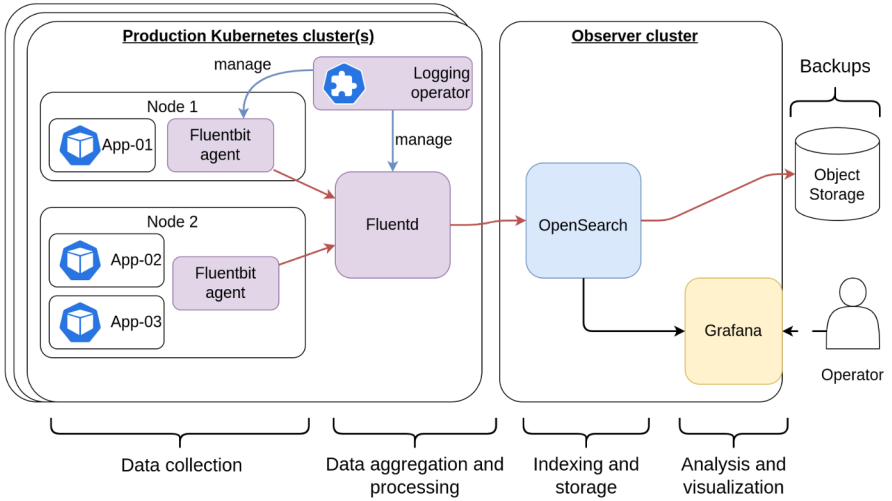
- API driven document-based database
- Highly Available and scalable
- Distributed architecture
- Advanced features like storage tiers, index templates and state management policies



Logs: Multi-cloud/cluster Case

We implemented an approach for all our production Kubernetes clusters to send their logs to a centralized Opensearch cluster, and define standards to increase the efficiency and performance of Opensearch

- Index naming convention for each **cluster**
- Field mappers, State management and Index Templates based on **application** type
- **logging-operator** applies config changes and log flows to clusters on commit
- **ClusterFlow** and **ClusterOutput** manage log routing to generic or special (e.g., ingress) indexes and templates



Example indexes:

- logs-ingress-cineca-eo4eu-ope-2024-02-27
- logs-generic-cineca-eo4eu-ope-2024-02-27

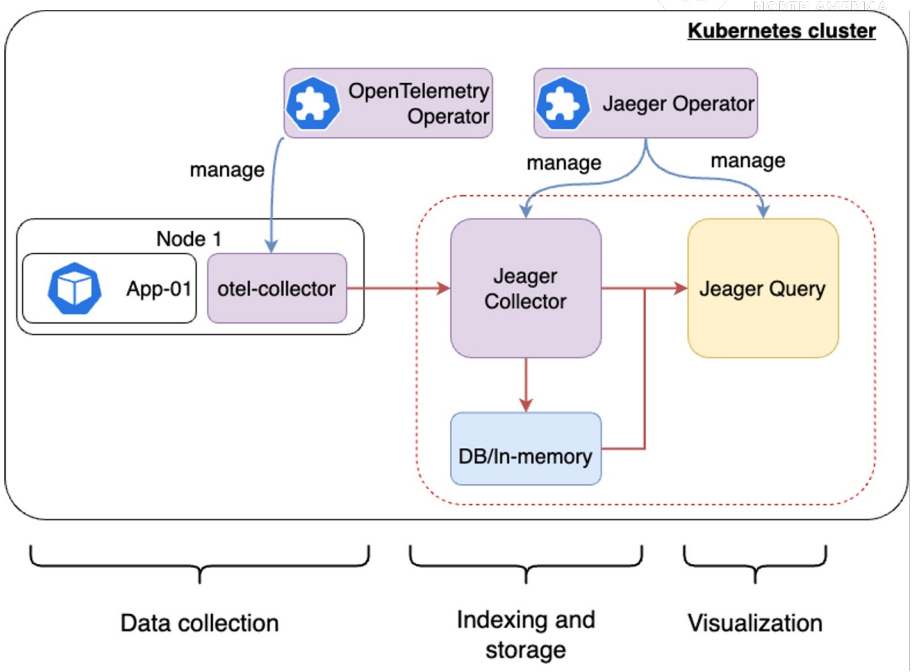
Traces: Single-cluster Case

OpenTelemetry (opentelemetry-operator/otel-collector)

- Vendor-agnostic proxy that can collect observability data
- De-facto standard
- Processors to enrich data (e.g. k8sattributes)
- Supports broad variety of backends
  - Jaeger
  - Prometheus/Thanos (prometheusremotewrite)

Jaeger (jaeger-operator/jaeger-all-in-one)

- Focused on distributed tracing in microservices
- Service dependency graphs
- Multiple storage backends
  - Cassandra
  - Opensearch/Elasticsearch



Traces: Multi-cloud/cluster Case

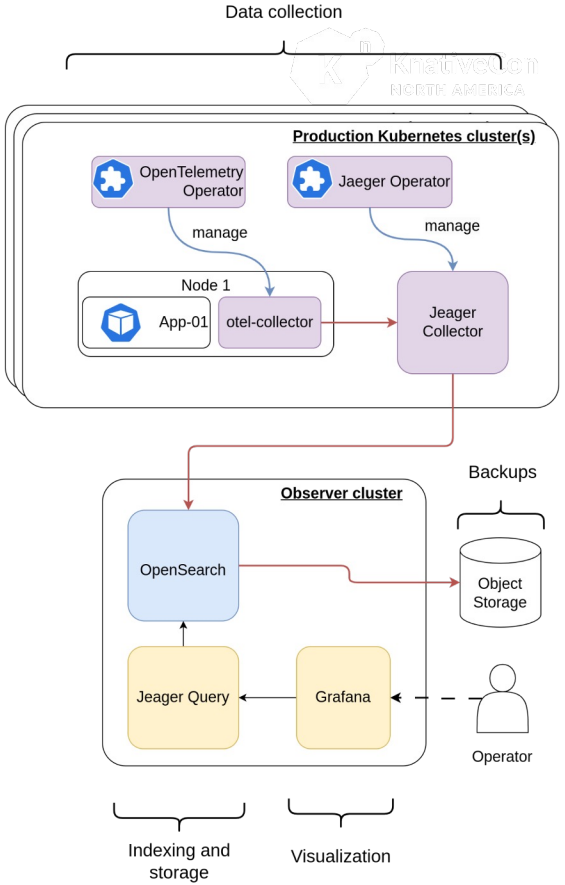
Observability Day EUROPE

OpenTelemetry (opentelemetry-operator/otel-collector)

- Configured to send traces to local Jaeger collector via OTLP exporter
- Using processors to enrich data
- ingress-nginx in each cluster is configured sent traces to otel-connector
  - "otel-service-name" = unique
  - "use-forwarded-headers"
  - Additional opentelemetry_attribute(s)

Jaeger (jaeger-operator/jaeger-production)

- Using Opensearch as a centralized storage backend for all collectors
- Traces are visualised by Grafana, in combination with the logs and metrics
- Each collector uses the same indices that Jaeger creates
  - Multi-tenancy is an ongoing discussion*
  - Using es.index-prefix=traces*



* https://github.com/jaegertracing/jaeger/issues/3881

Our Multi-cloud Journey

Challenges

- How to integrate several tools?
- Deploy and manage tools at scale (requires automation)
- Multi tenancy for heterogeneous teams access (e.g., RBAC)
- Missing useful documentation (we contributed to this*)
- Networking solution (e.g., ingress or cross-cluster connectivity and service discovery)

Design principles

- Open-source and open-license ecosystem
- Automation through GitOps and Infrastructure as Code (IaC)
- Kubernetes-first approach:
  - Official operators as first choice
  - Well-supported Helm charts
  - Custom solutions when needed (e.g., Grafana Dashboards and Alerts)

We started with GitLab CI and Terraform, then replaced with ad-hoc solutions and GitOps tools.

Contributions:

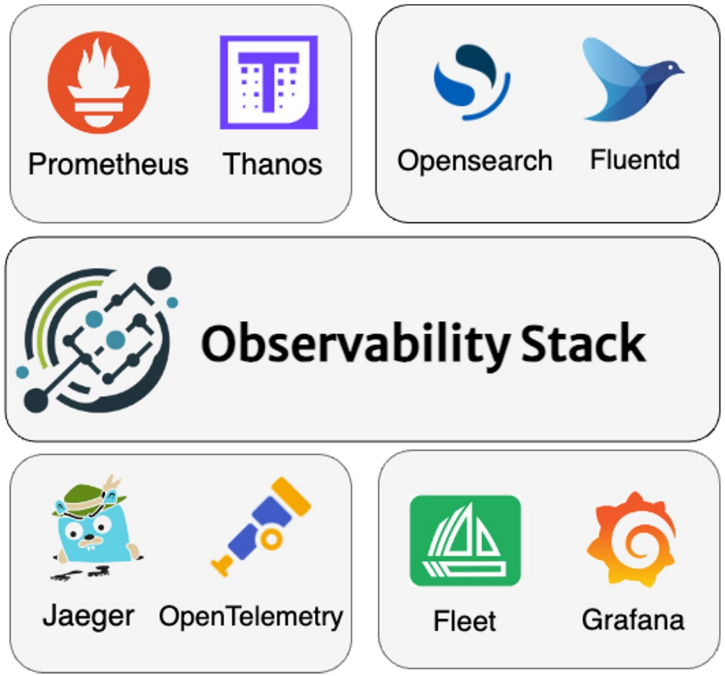
- *Documentation: PVC and Volumes example · Issue #1391 · grafana/grafana-operator*
- *Update rbac-for-monitoring.md by armagankaratosun · Pull Request #1104 · rancher/rancher-docs*

Introducing Observability Stack

Observability Day EUROPE

By leveraging the available open-source ecosystem, we come up with the idea of Observability Stack

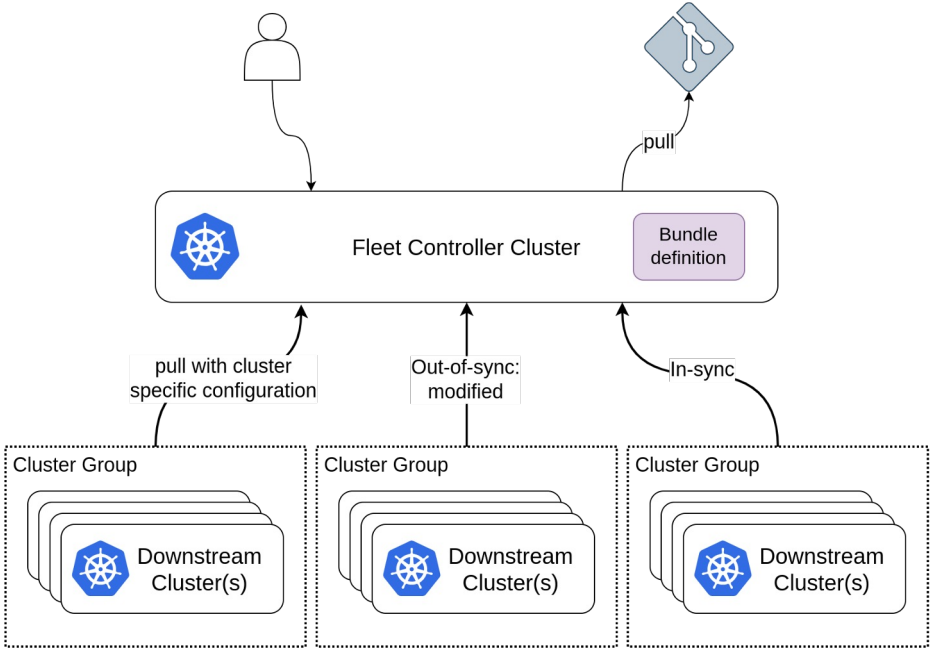
- **Not a product** - an umbrella project, to provide a platform agnostic and flexible observability toolkit for administrators and developers to build their solutions tailored to their unique requirements.
- **Automated Multi-cloud deployments** - by embracing GitOps as the 4th Pillar of Observability, we automatically configure clusters in our multi-cloud infrastructure to transmit metrics, logs, and tracing data to a central "**observer**" cluster, ensuring simplified deployment, day 2 operations and consistency.



GitOps as the 4th Pillar

Observability stack uses **Fleet (by Rancher)** for the distribution of its components and the management of their lifecycle.

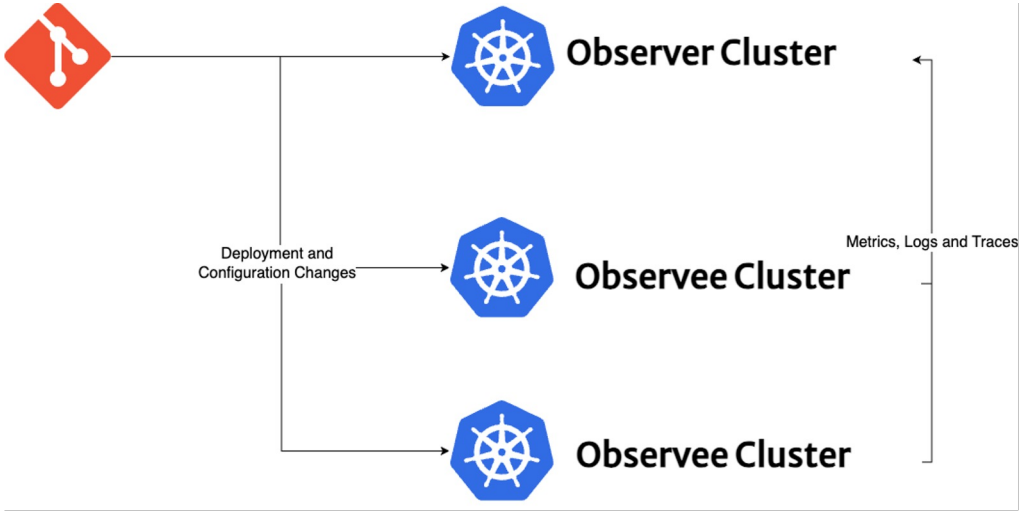
- **Cluster Labels:** Each cluster registered as a downstream cluster to Fleet Manager can be labelled to form groups
- **Cluster Group:** Clusters connected to the same Fleet Manager can be organized into groups with a ***matchLabels*** selector
- **Observability Stack** uses this mechanism to form Cluster Groups with labels:
  - `observability-role: observer`
  - `observability-role: observee`



GitOps as the 4th Pillar

By adopting a logical separation of "**observee**" and "**observer**" cluster labels and cluster groups, each cluster is configured to transmit its observability data, ensuring that the **observer** cluster has a centralized dataset to analyze.

- **Simplifies the deployment** of the Observability Stack to the downstream clusters
- **Auto-enroll** newly provisioned cluster as Observee
- **Automates the configuration changes** (e.g., adding new ClusterFlow) and version updates
- **Ensures consistency** within the multi-cloud/cluster setup
- **Plug-and-play architecture -** components of the Observability Stack can be changed easily



EO4EU Observability Platform

The EO4EU Observability Platform uses **Observability Stack** to observe the entire multi-cloud/cluster from infrastructure to application level and it leverages on a **observee** and **observer cluster** role subdivision.

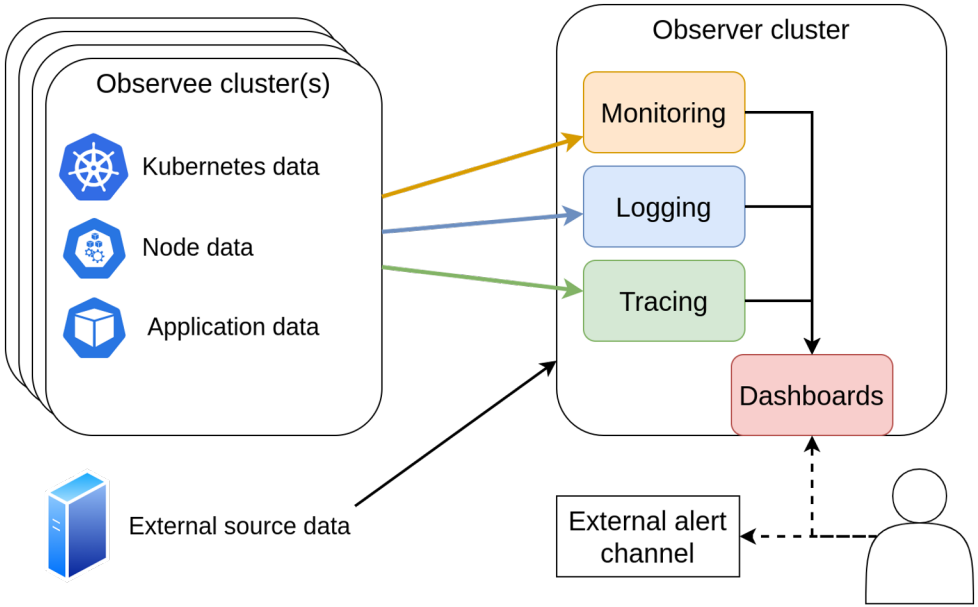
The **Observer** is the central cluster which allows to query, collect and visualize all data coming from **Observee** clusters which host all the EO4EU platform services.

Features:

- Scalable
- Highly available
- Full view on all sources
- Single access point for operators and developers

Tools:

- Monitoring (Prometheus, Thanos)
- Logging (Fluentbit, Fluentd)
- Tracing (OpenTelemetry, Jeager)
- Indexing (OpenSearch)
- Visualization (Grafana)



Live Demo

Let's hope it will work :)

If not we have backup pictures

Moving Forward

Join us in overcoming the challenges on multi-cloud observability at

observability-stack.io

observability-stack/observability-stack

Observability Day
EUROPE

Thank you!

Please leave feedback

EO4EU
Earth Observation Data & Service Accessibility

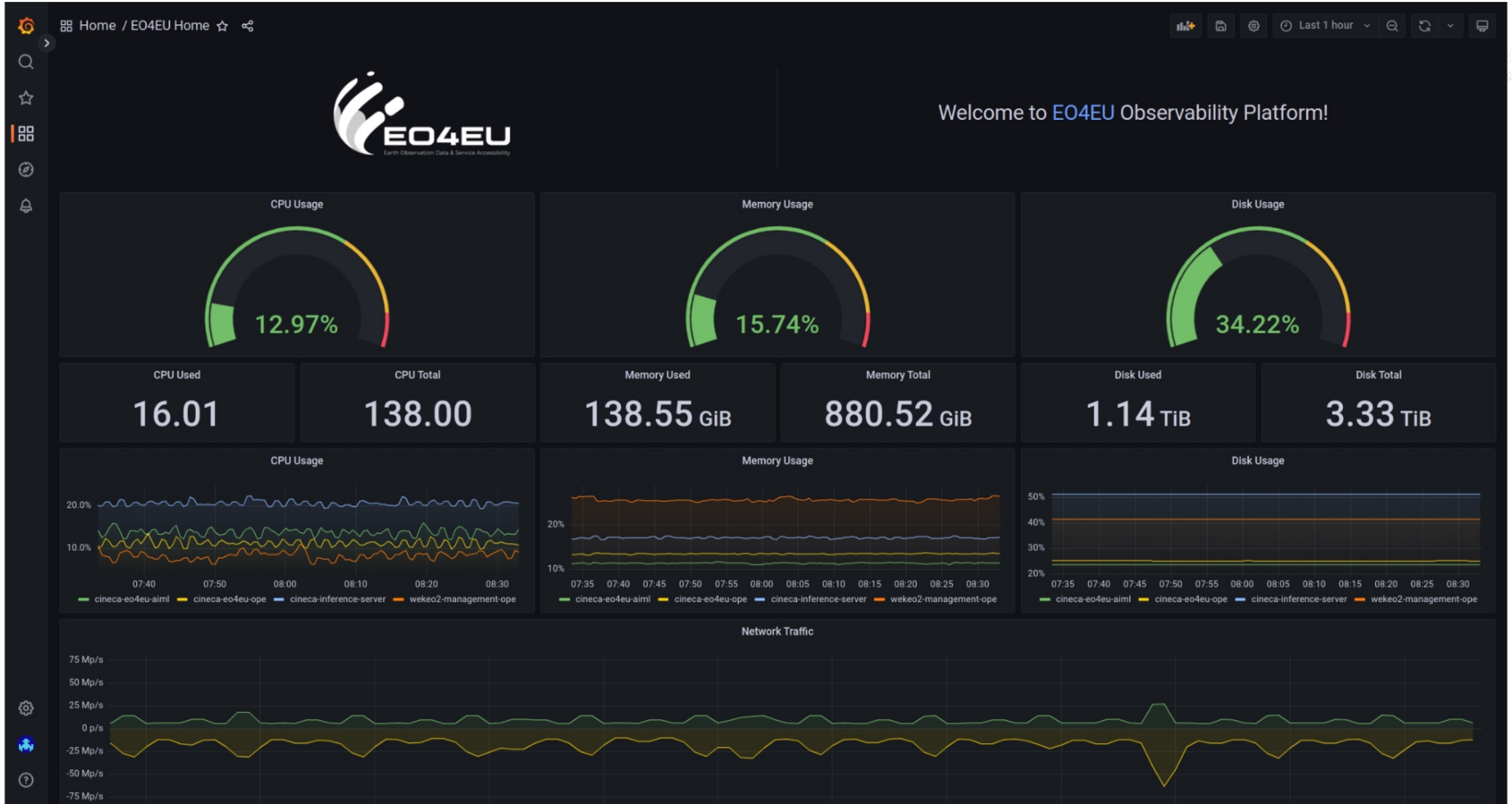
https://www.eo4eu.eu

Funded by
the European Union

Work funded by the EU Horizon Europe grant 101060784 (EO4EU)

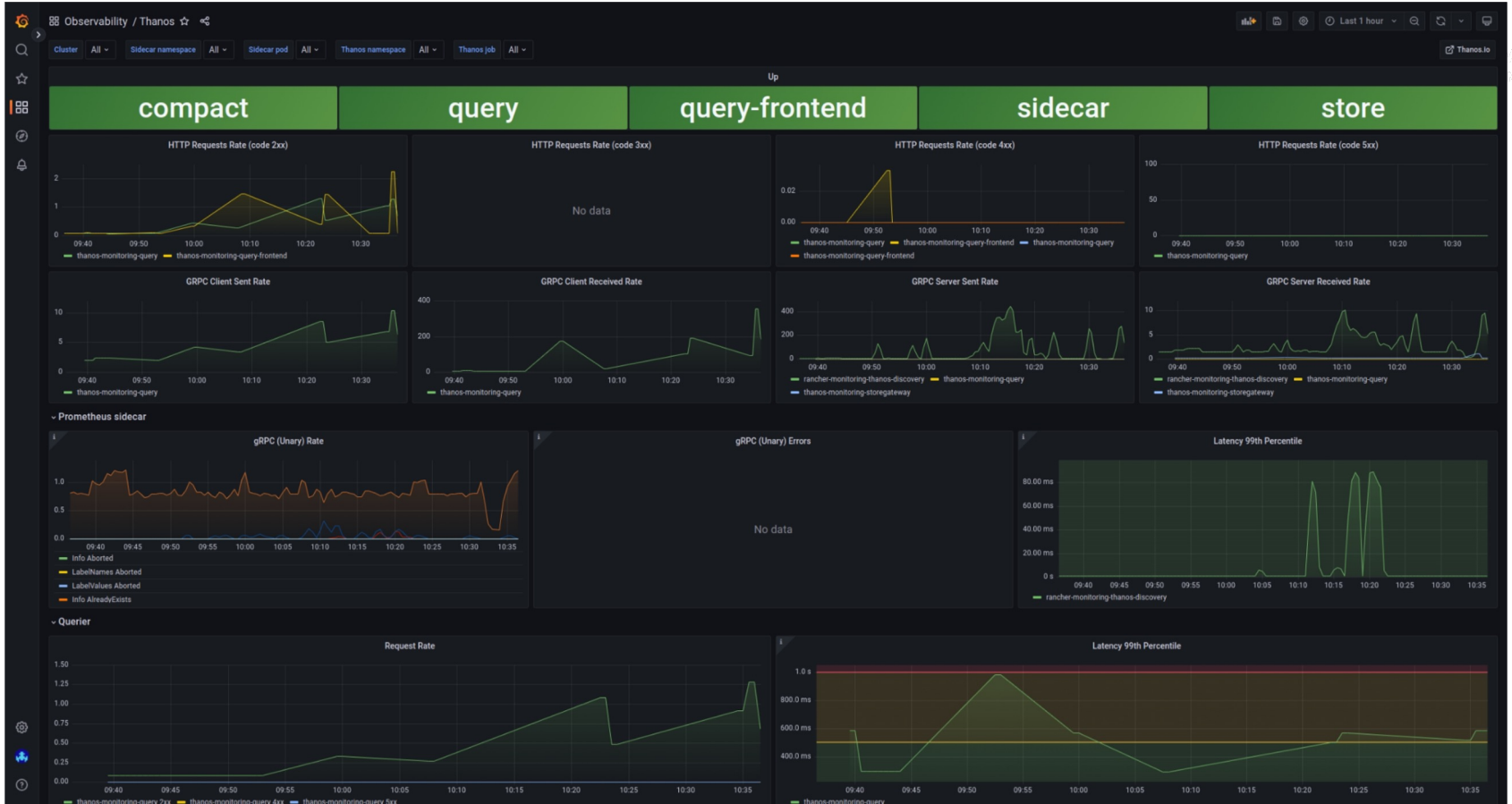
Backup Demo

Observability Day EUROPE

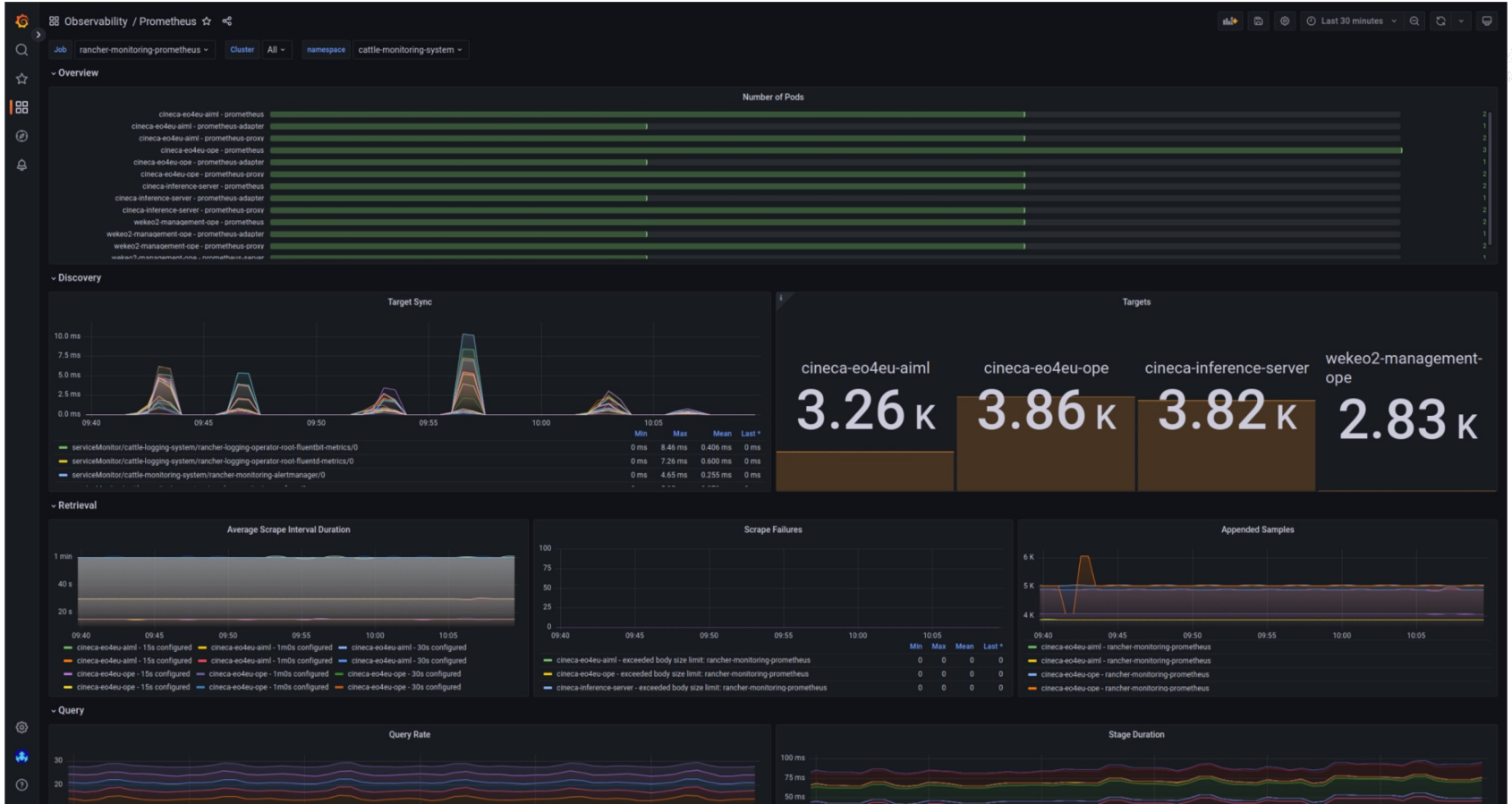


Dashboards: Metrics

Observability Day EUROPE

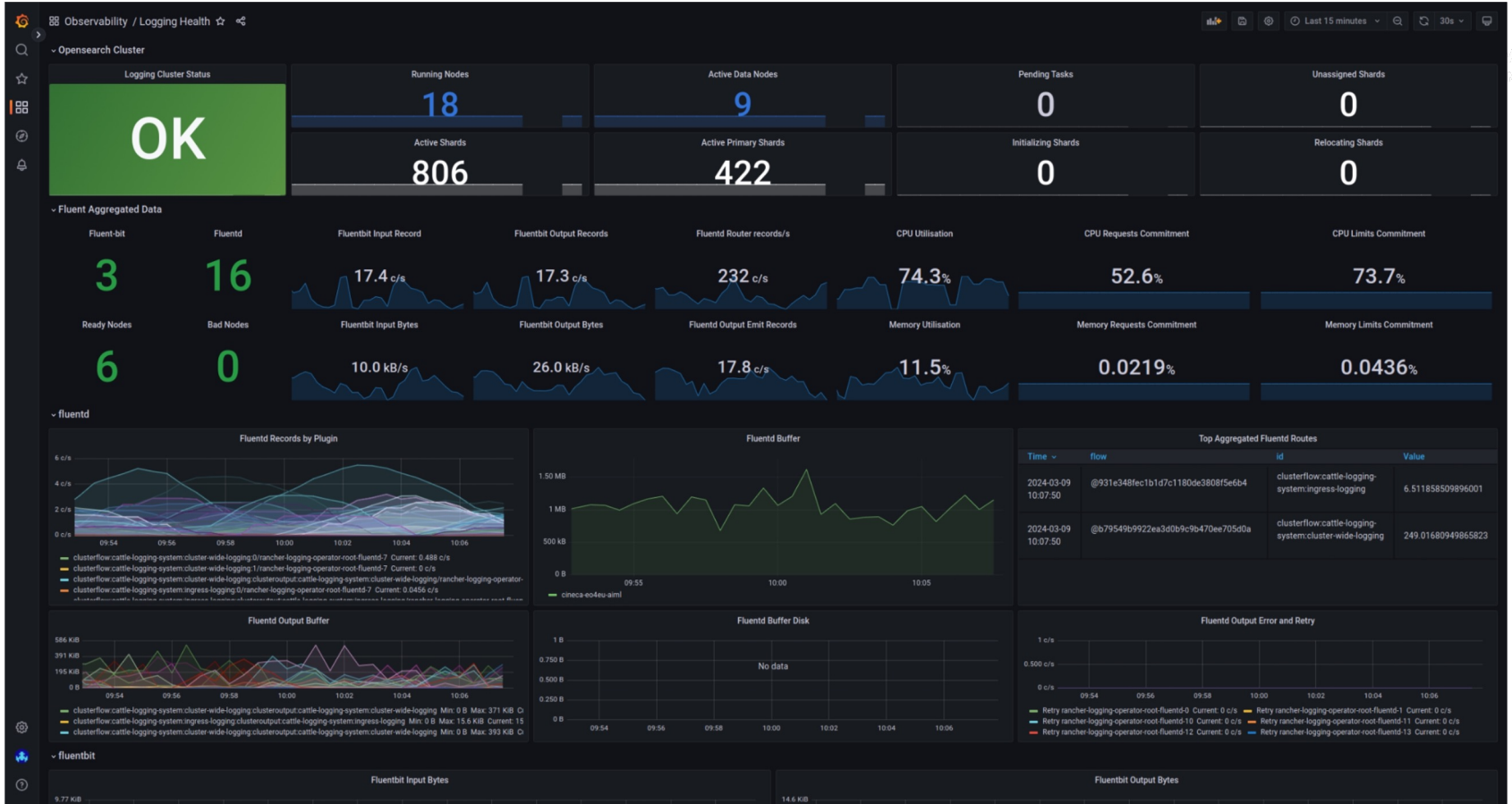


Dashboards: Metrics



Dashboards: Metrics

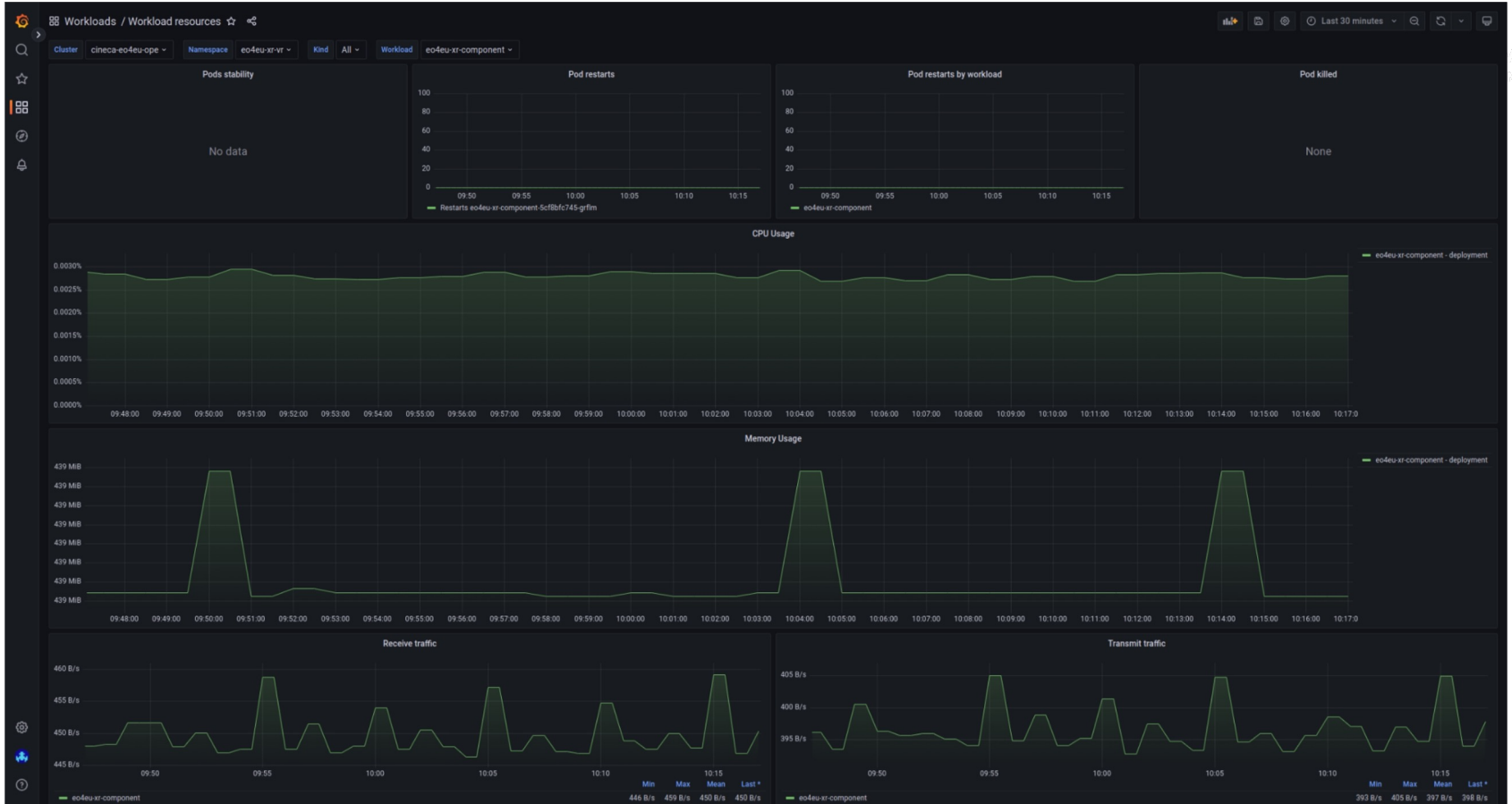
Dashboards: Metrics



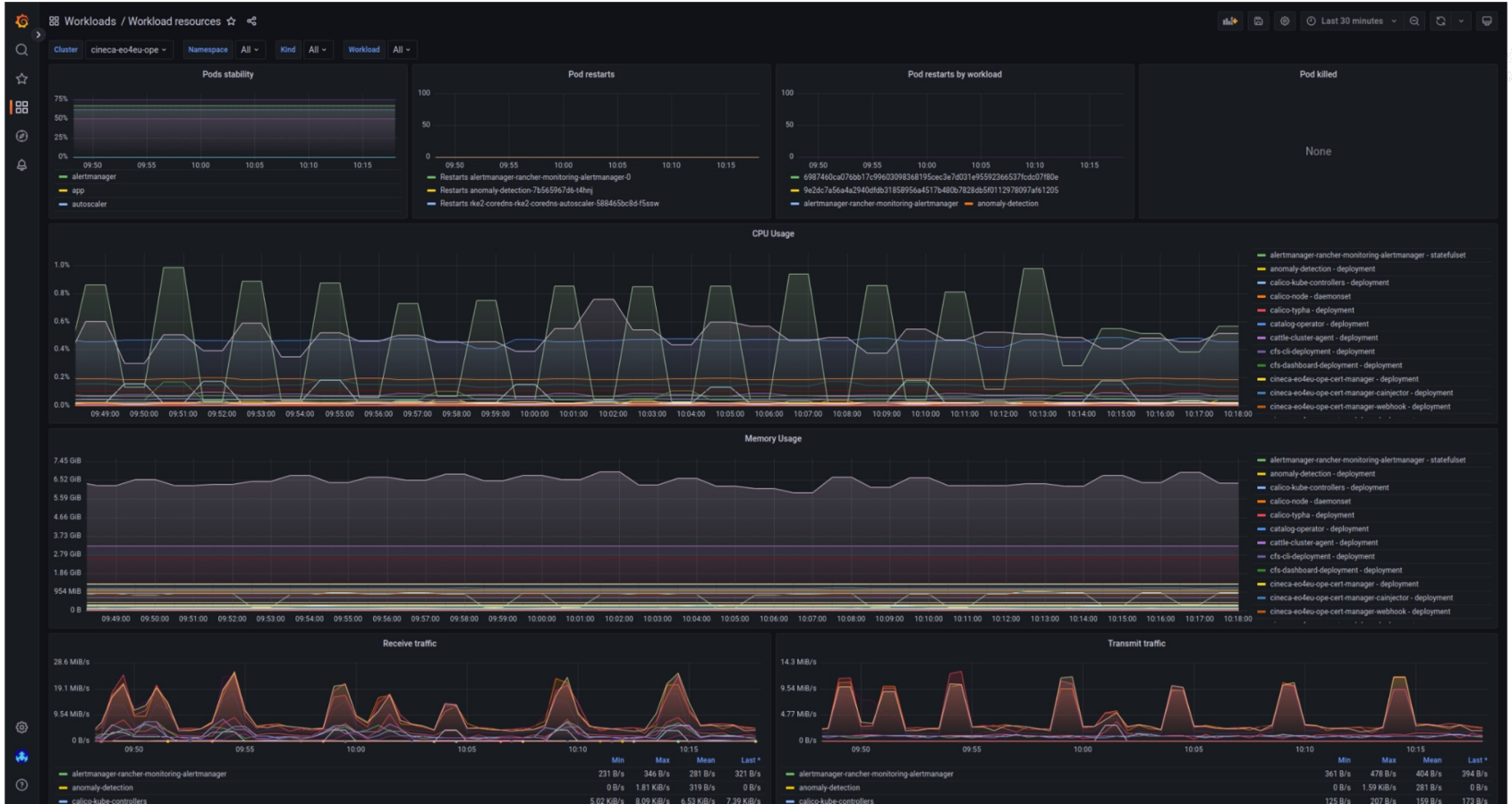
Dashboards: Metrics

Observability Day EUROPE

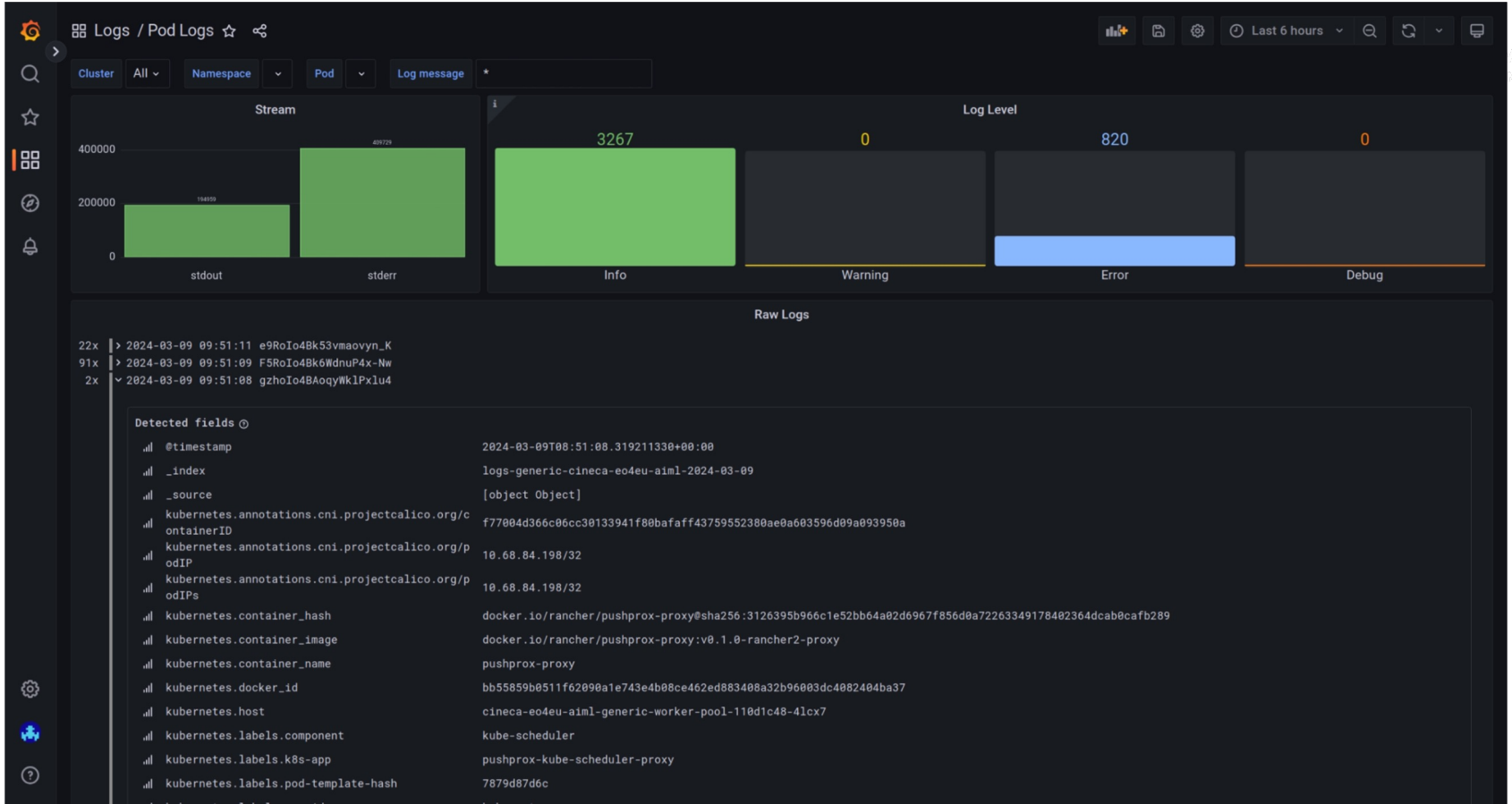
Dashboards: Metrics



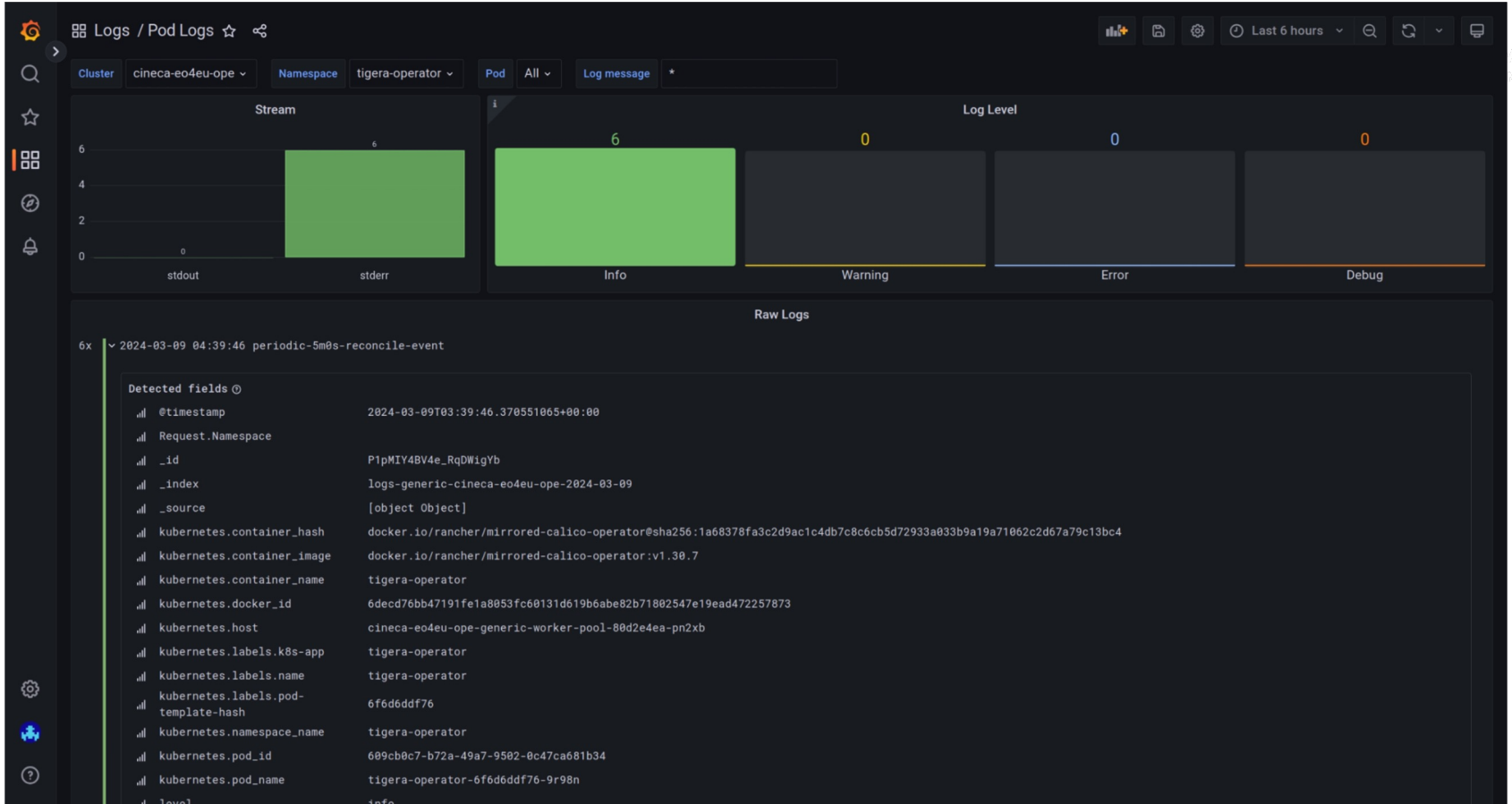
Dashboards: Metrics



Dashboards: Logs



Dashboards: Logs



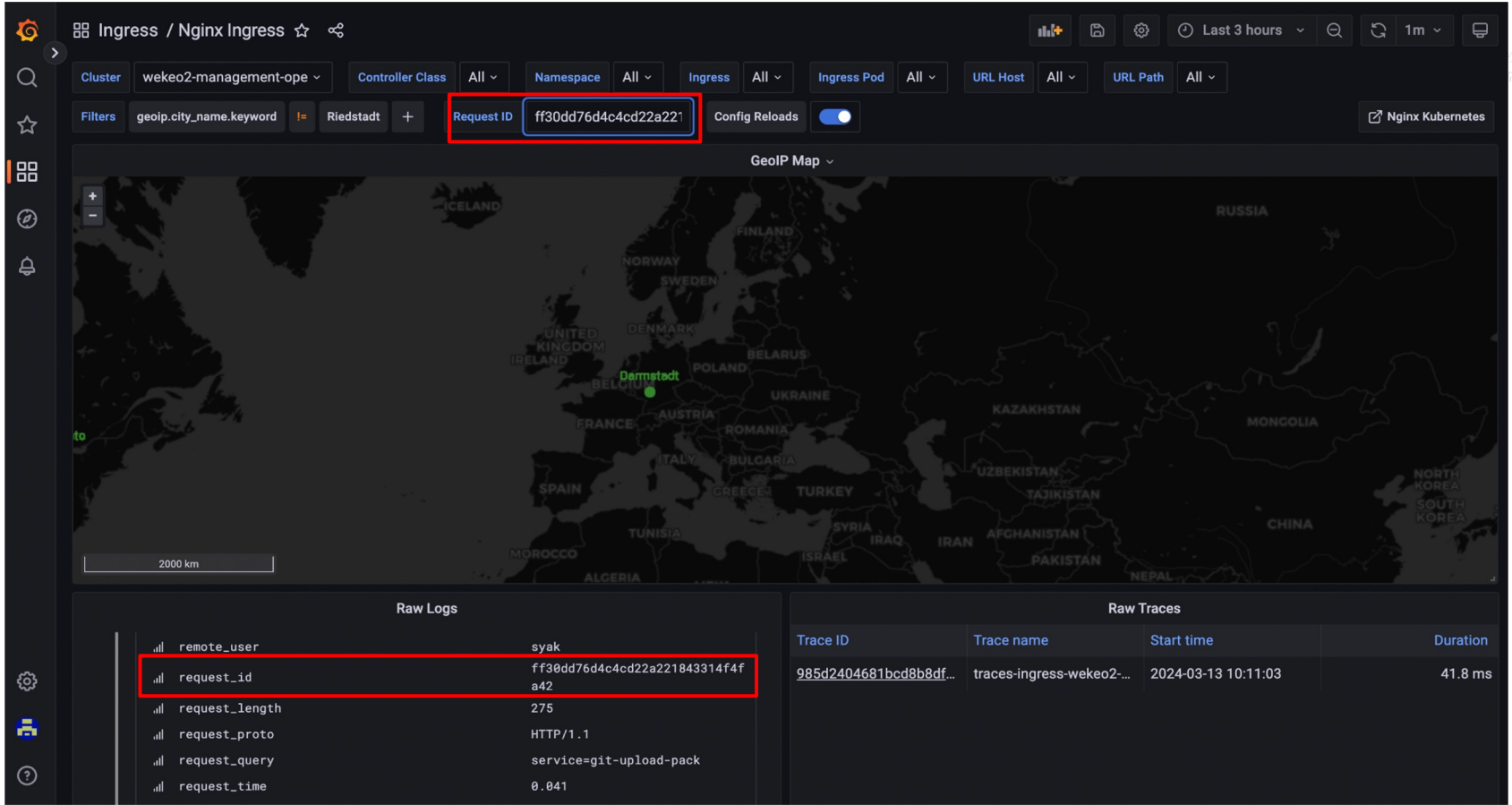
Dashboards: Logs & Traces

Observability Day EUROPE

Select log request ID to select only traces related to logs

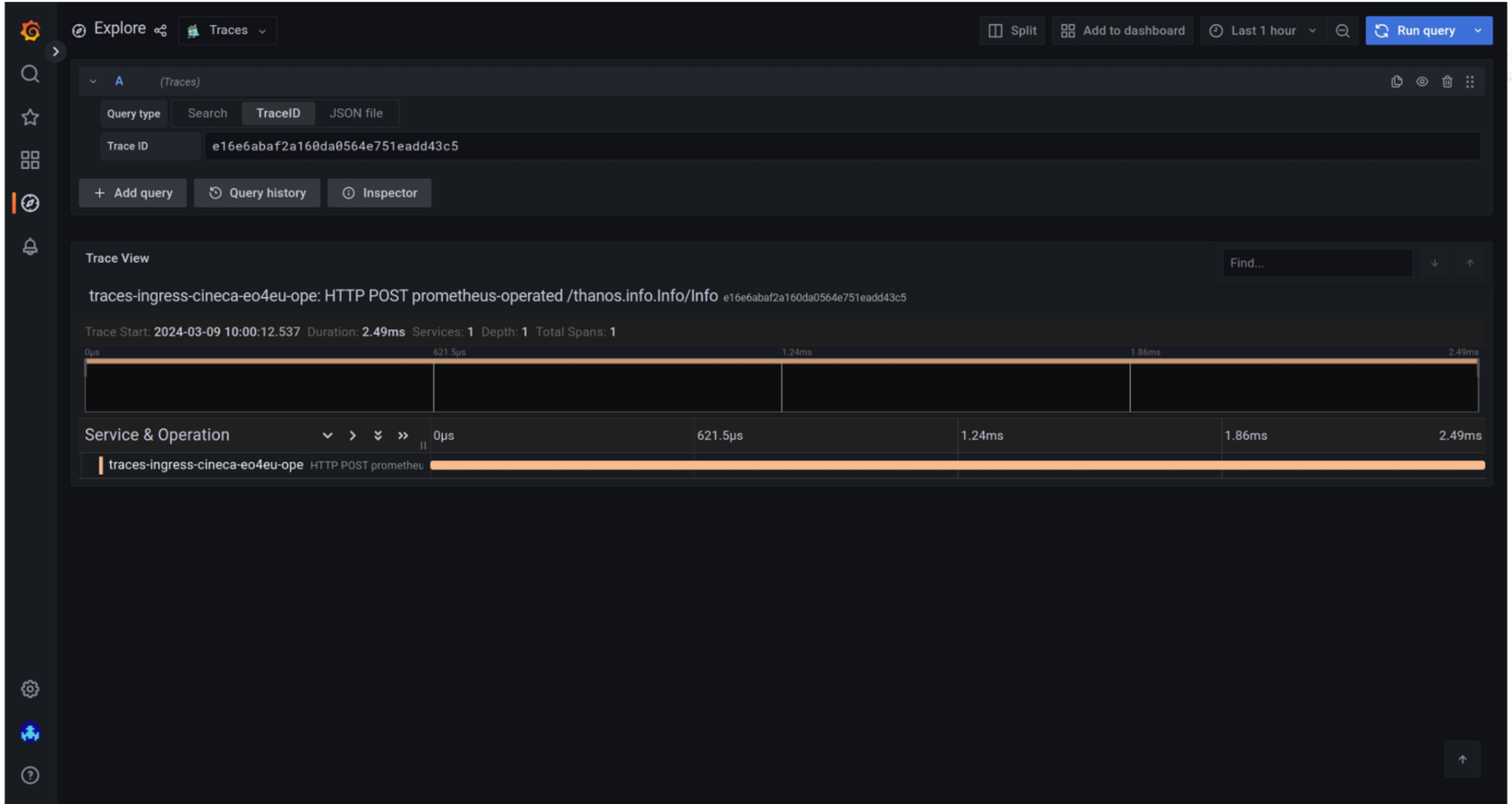
KubeCon NORTH AMERICA

Dashboards: logs & traces



Dashboards: Traces

Observability Day EUROPE



Dashboards: Tracing view

Observability Day EUROPE