



KubeCon



CloudNativeCon

Europe 2026

To upstream or not? Why becoming the maintainer of your dependencies matters

Christos Markou, Software Engineer @ Elastic



whoami



Christos Markou

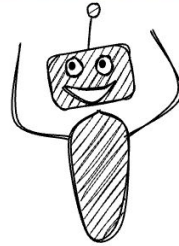
- Principal Software Engineer @ Elastic
- OpenTelemetry Collector Contrib Maintainer
- Semantic Conventions Approver (system, k8s, containers)
- CNCF Ambassador

Why this session

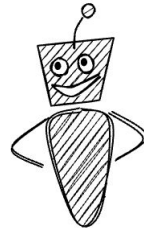
✓ KubeCon + OSS = ❤️



People of OSS



Should we become these
people and how?





KubeCon



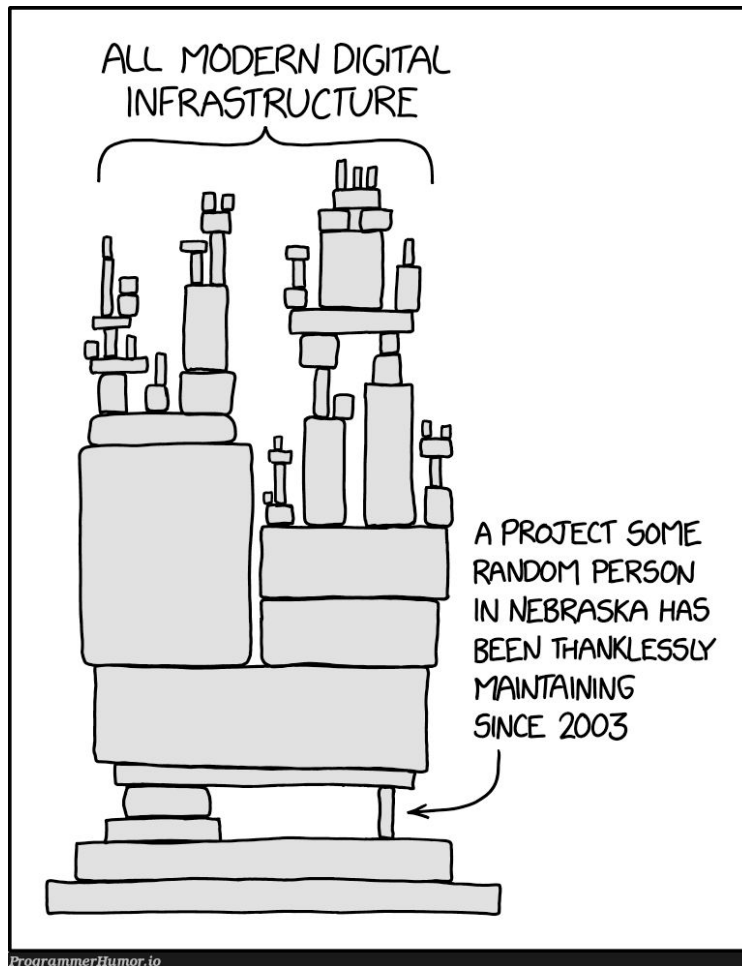
CloudNativeCon

Europe 2026

OpenSource importance



OSS importance



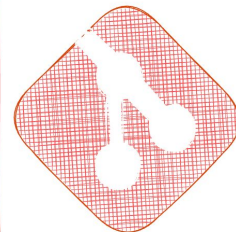
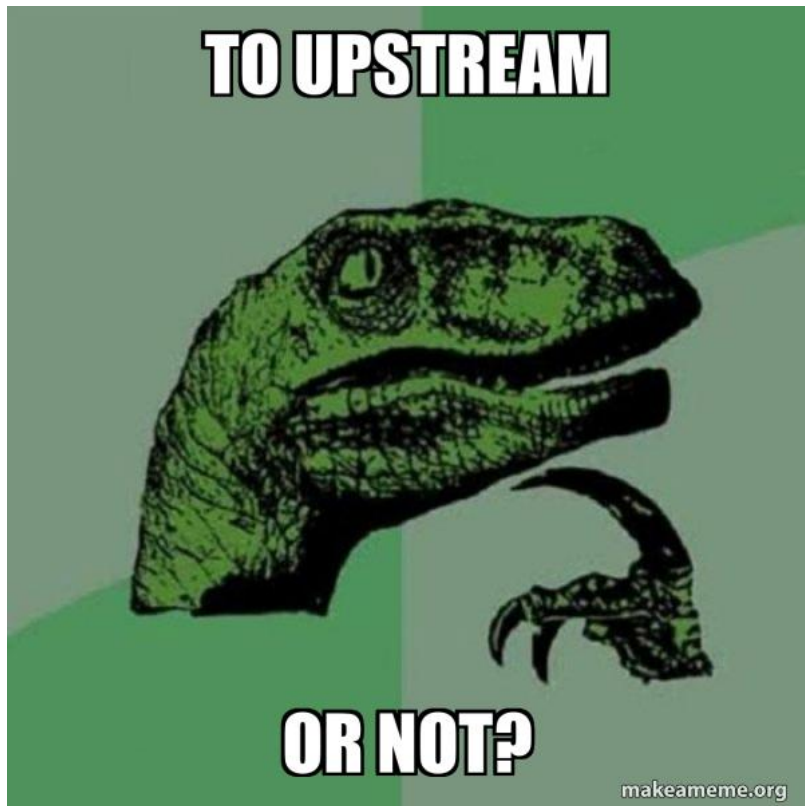
When things go south

Ingress NGINX Retirement: What You Need to Know

By **Tabitha Sable (Kubernetes SRC)** | Wednesday, November 12, 2025

To prioritize the safety and security of the ecosystem, Kubernetes SIG Network and the Security Response Committee are announcing the upcoming retirement of [Ingress NGINX](#). Best-effort maintenance will continue until March 2026. Afterward, there will be no further releases, no bugfixes, and no updates to resolve any security vulnerabilities that may be discovered. **Existing deployments of Ingress NGINX will continue to function and installation artifacts will remain available.**

<https://www.kubernetes.dev/blog/2025/11/12/ingress-nginx-retirement/>



upstream: source repository
and project where contributions
happen and releases are made



KubeCon



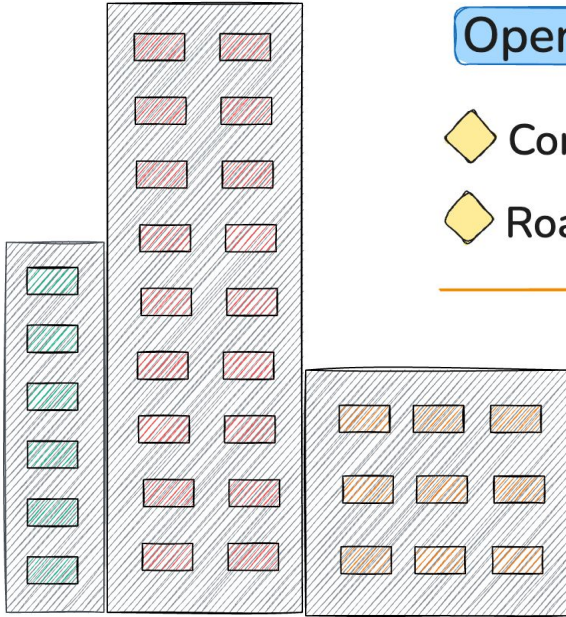
CloudNativeCon

Europe 2026

Stakeholders and dynamics of OSS



Vendors/companies

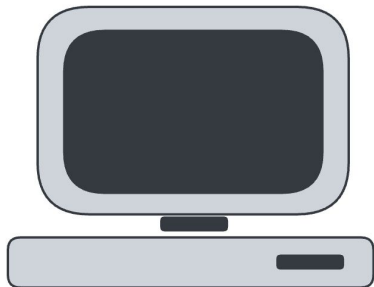


Open source ecosystems are powered by companies

- ◆ Core maintainers are often employed by vendors
 - ◆ Roadmaps and priorities are shaped by corporate investment
-

- Companies want to monetize their investment (and that's fine)
- It doesn't make open source less open
- Governance, balance, and transparency are essential

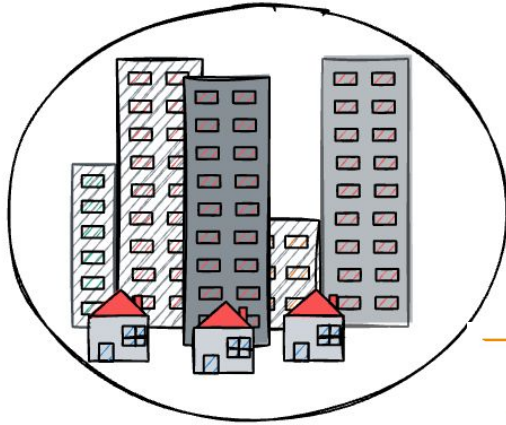
OSS maintainers



Open source maintainers are key stakeholders

- ◆ Carry long-term technical ownership
 - ◆ Review, approve, and reject changes
 - ◆ Maintain project standards and quality
-
- Responsibility is formalized through Technical Committees, Governance Committees, or SIGs
 - Doesn't make maintainers gatekeepers
 - Makes clear roles, fair processes, and accountability essential.

The community users



User community is a core stakeholder

- ◆ adopt the software in real-world systems
- ◆ surface bugs, edge cases, and limitations
- ◆ bring feedback, feature requests, and pressure to evolve

-
- users don't control the roadmap
 - their needs and trust define the project's relevance

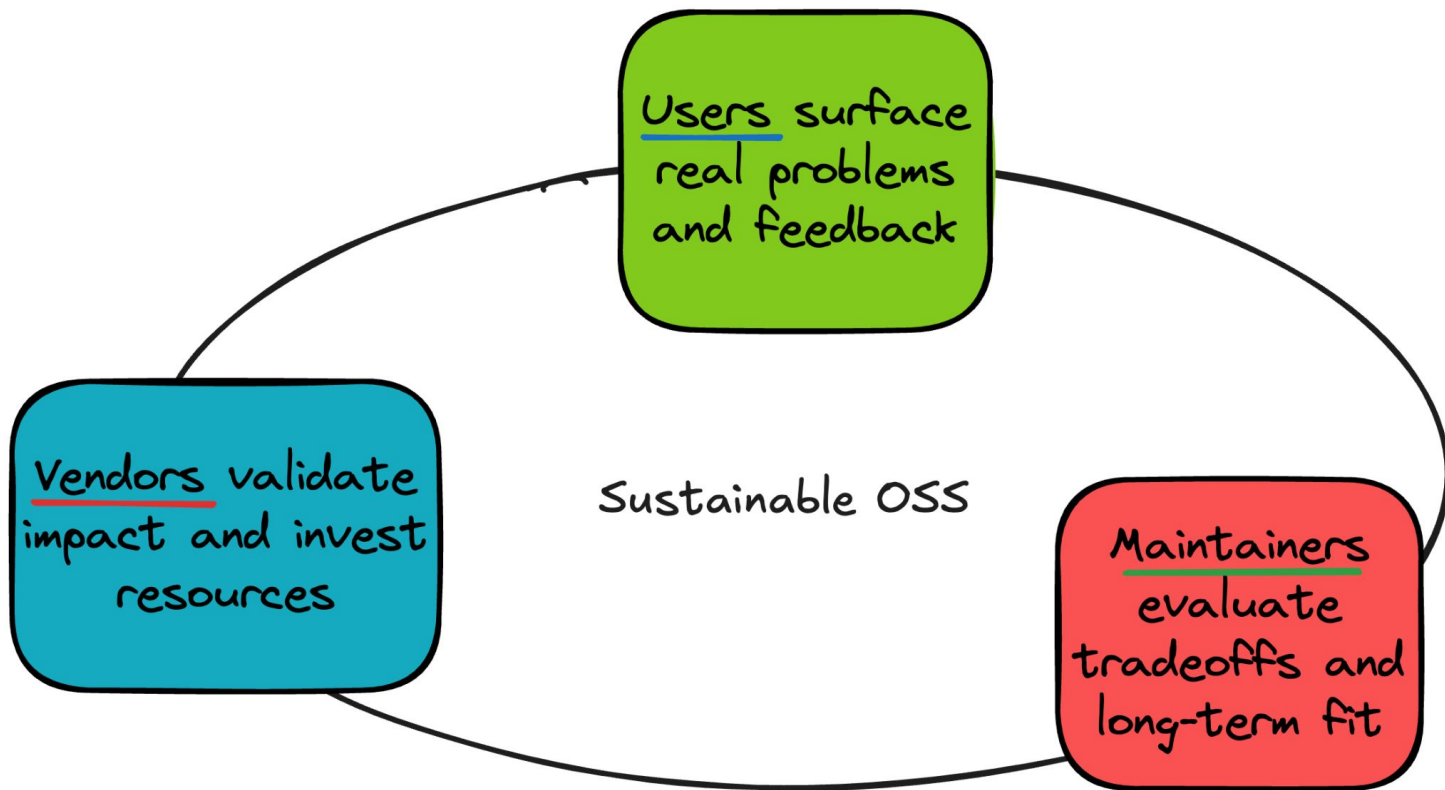


KubeCon



CloudNativeCon

Europe 2026





KubeCon



CloudNativeCon

Europe 2026



KubeCon



CloudNativeCon

Europe 2026

OpenSource O11y and OpenTelemetry



What's OpenTelemetry



- ◆ 2nd largest project in the CNCF
- ◆ an OSS standard for telemetry (metrics, traces, logs etc)
- ◆ vendor-neutral telemetry collection
- ◆ vast ecosystem of tooling to enable powerful instrumentation

OTel Collector architecture



vendor neutral telemetry agent



ingests telemetry through receivers



transform & enrich with processors



send to any backend via exporters



KubeCon



CloudNativeCon

Europe 2026

OTel Collector distributions

The Collector is **vendor-neutral** but vendors provide their own distributions to simplify adoption and add value

Examples:

ADOT (AWS Distro for OpenTelemetry) – AWS's distribution

EDOT (Elastic Distro for OpenTelemetry) – Elastic's distribution

DDOT (Datadog Distro for OpenTelemetry) – Datadog's distribution

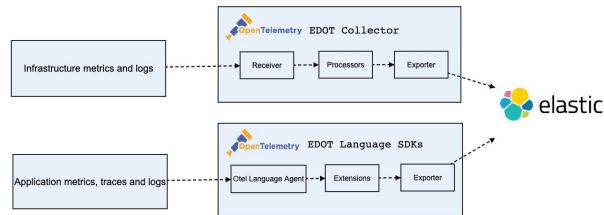
OTel Collector distributions

Why distributions exist

- ◆ Pre-packaged for easier deployment and scaling
- ◆ Include vendor-specific receivers, exporters, or processors
- ◆ Provide additional operational tooling, monitoring, or documentation
- ◆ Allow faster adoption/support without modifying the upstream Collector

What is EDOT Collector

Elastic Distributions for OpenTelemetry



A downstream distribution of the upstream Collector

imports upstream components

upstream components as direct dependencies of what we ship

EDOT follows Elastic's release cadence, not the upstream one

Elastic-specific components/defaults baked into the distribution



KubeCon



CloudNativeCon

Europe 2026

EDOT and k8s Events (a real story)



EDOT and k8s Events



k8sevents receiver

Connects to the Kubernetes API

Gets Kubernetes events as logs

Should only run once in a cluster
(singleton required)

Adds k8s metadata about related
objects

```
{  
  "Body": "Pulling image nginx",  
  "k8s.event.start_time": "2025-09-08T06:20:41Z",  
  "k8s.node.name": "kind-control-plane",  
}
```



KubeCon



CloudNativeCon

Europe 2026

EDOT and k8s Events



k8sobjects receiver

Connects to the Kubernetes API

Watches / Pulls Kubernetes Objects

Should only run once in a cluster
(singleton required)

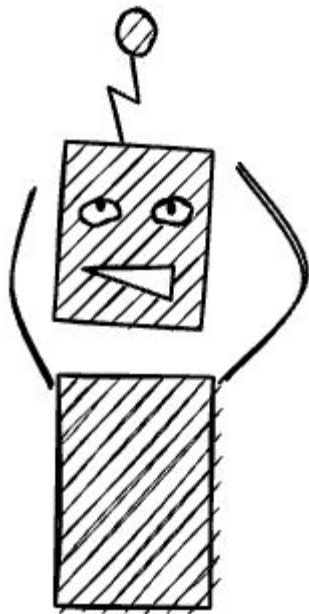
Emits raw objects as logs

```
{
  "apiVersion": "v1",
  "kind": "Pod",
  "metadata": {
    "creationTimestamp": "2025-11-03T08:29:23Z",
    "generateName": "daemonset-opentelemetry-collector-agent-",
    "labels": {
      "app.kubernetes.io/instance": "daemonset",
      "app.kubernetes.io/name": "opentelemetry-collector",
      "component": "agent-collector",
      "controller-revision-hash": "54d7d685bc",
    },
    "name": "daemonset-opentelemetry-collector-agent-jnjgs",
    "namespace": "default",
    "ownerReferences": [
      {
        "apiVersion": "apps/v1",
        "blockOwnerDeletion": true,
        "controller": true,
        "kind": "DaemonSet",
        "name": "daemonset-opentelemetry-collector-agent",
        "uid": "d303b30c-2a2a-4924-a0bc-6c0f3aaa02c4"
      }
    ],
    "resourceVersion": "47877",
    "uid": "ade8de73-fc3f-4091-9c6d-f7c34ba49754"
  },
  "spec": { ... },
  ...
}
```

K8sObjects for events

```
1 receivers:
2   k8sobjects:
3     objects:
4       - name: events
5         mode: "watch"
6         group: "events.k8s.io"
7         exclude_watch_type:
8           - "DELETED"
```

Which one to use?



K8sevents marked for deprecation

[receiver/k8sevents] Deprecate the k8seventsreceiver #24242

Closed

TylerHelmuth opened on Jul 12, 2023 · edited by TylerHelmuth

Component(s)

receiver/k8sevents, receiver/k8sobjects

Describe the issue you're reporting

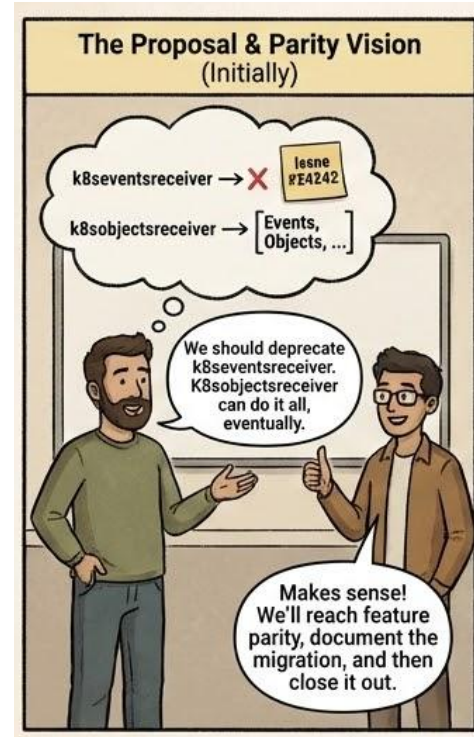
With the introduction of the [k8sobjectsreceiver](#) the purpose of the k8seventsreceiver is now handle by a more generic component. I propose we deprecate the k8seventsreceiver in favor of the k8sobjectsreceiver.

Work to do:

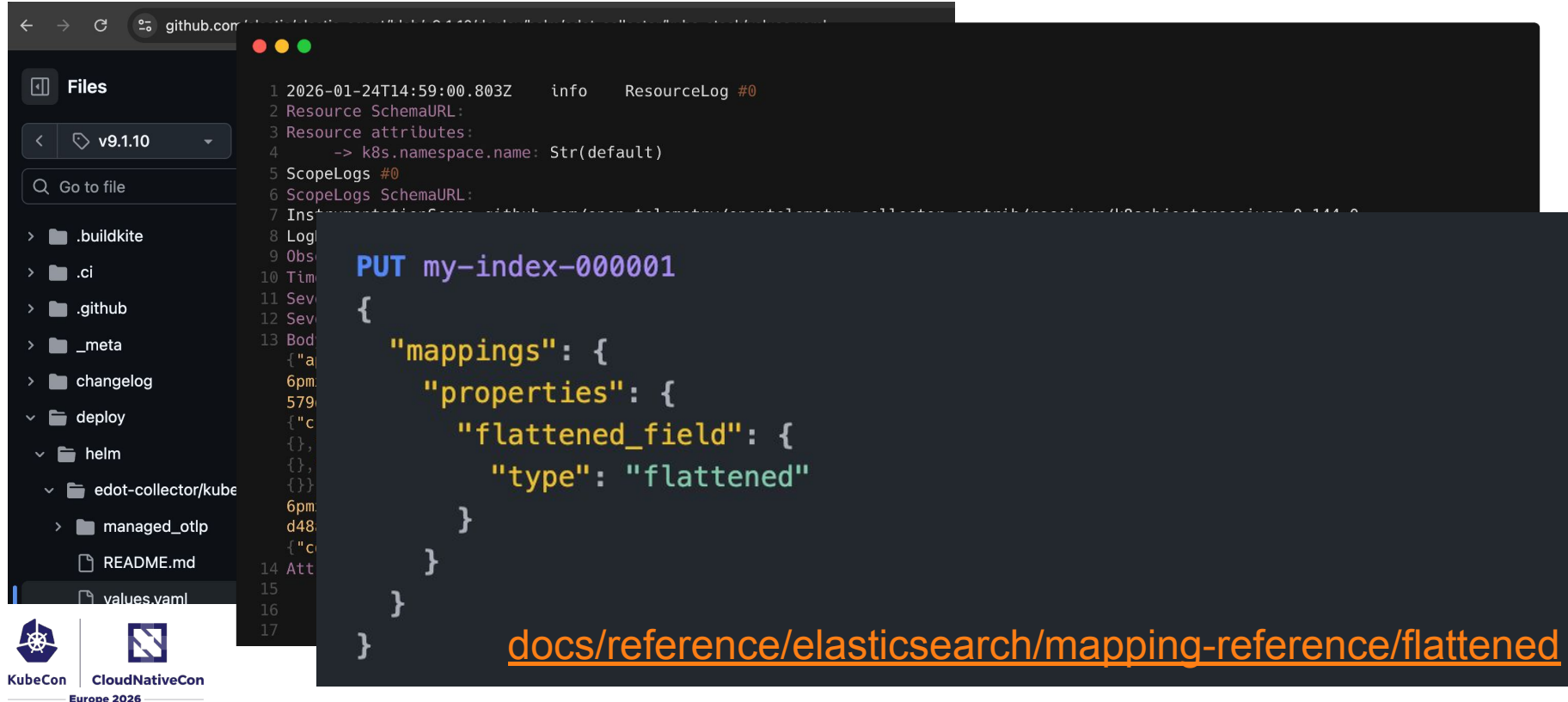
- ☒ Feature gap analysis to ensure all the functionality in k8seventsreceiver is handled by the k8sobjectsreceiver
- ☐ Fill any gaps
- ☐ Provide a migration path with a configuration that would produce the same data using k8sobjectsreceiver
- ☐ Follow deprecation process

Create sub-issue

TylerHelmuth added priority:p2 discussion needed receiver/k8sevents receiver/k8sobjects on Jul 12, 2023



EDOT is shipped with k8sobjects



The image shows a GitHub repository interface on the left and a terminal window on the right. The GitHub interface displays the file structure of the 'edot-collector/kube' directory, with 'values.yaml' selected. The terminal window shows logs from 'ResourceLog' and a PUT request to 'my-index-000001' with a JSON body containing a flattened field.

```
1 2026-01-24T14:59:00.803Z    info    ResourceLog #0
2 Resource SchemaURL:
3 Resource attributes:
4   -> k8s.namespace.name: Str(default)
5 ScopeLogs #0
6 ScopeLogs SchemaURL:
7 InstrumentationScope github.com/elastic/edot-collector-contrib/instrumentation/otelcollector/2.14.0
8 Log
9 Obs
10 Tim
11 Sev
12 Sev
13 Bod
   {"a
   6pm
   579
   {"c
   {},
   {},
   {}
   6pm
   d48
   {"c
14 Att
15
16
17
```

```
PUT my-index-000001
{
  "mappings": {
    "properties": {
      "flattened_field": {
        "type": "flattened"
      }
    }
  }
}
```

[docs/reference/elasticsearch/mapping-reference/flattened](https://docs.elasticsearch.com/mapping-reference/flattened)

KubeCon CloudNativeCon Europe 2026

The Long Silence (Many Months Later...)



The Community Queries (The Questioning Phase)



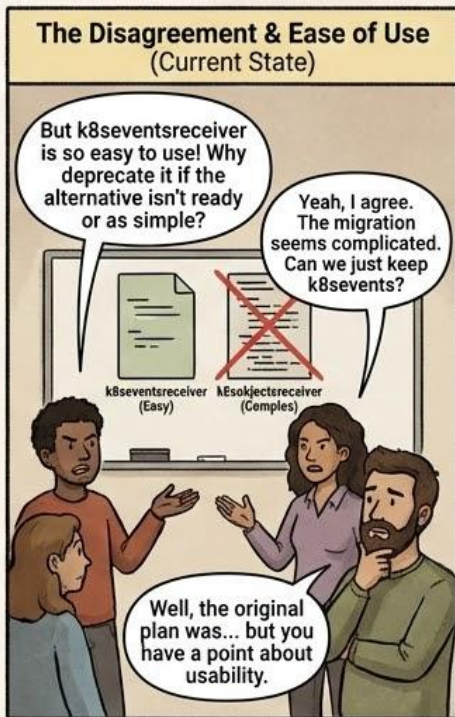
KubeCon



CloudNativeCon

Europe 2026

Usage of k8sobjects is not ideal



🔒 Closed Multiple null values when collecting events #39922

Description

The default events configuration produces a lot of noise with null fields that doesn't seem valuable

The k8sreceiver doesn't produce any noise and collects a clean structure

Steps to Reproduce

Put receiver configuration

```
k8sobjects:
  objects:
    - name: events
      mode: watch
```

Expected Result

Clean structure with the defined fields only

Actual Result

► Noisy structure

Usage of k8sobjects is not ideal

```
1 processors:
2   transform/events:
3     error_mode: ignore
4     log_statements:
5       - context: log
6         statements:
7           - set(resource.attributes["k8s.node.name"], body["object"]["source"]["host"])
8
9           - set(resource.attributes["k8s.object.kind"], body["object"]["involvedObject"]["kind"])
10          - set(resource.attributes["k8s.object.name"], body["object"]["involvedObject"]["name"])
11          - set(resource.attributes["k8s.object.uid"], body["object"]["involvedObject"]["uid"])
12          - set(resource.attributes["k8s.object.fieldpath"], body["object"]["involvedObject"]["fieldPath"])
13          - set(resource.attributes["k8s.object.api_version"], body["object"]["involvedObject"]["apiVersion"])
14          - set(resource.attributes["k8s.object.resource_version"], body["object"]["involvedObject"]["resourceVersion"])
15
16          - set(log.cache["time"], body["object"]["firstTimestamp"]) where body["object"]["firstTimestamp"] != nil
17          - set(log.cache["time"], body["object"]["lastTimestamp"]) where body["object"]["lastTimestamp"] != nil
18          - set(log.cache["time"], body["object"]["eventTime"]) where body["object"]["eventTime"] != nil
19          - set(log.time, Time(attributes["time"], "%Y-%m-%dT%H:%M:%SZ"))
20          - delete_key(log.cache, "time")
21
22          - set(attributes["k8s.event.reason"], body["object"]["reason"])
23          - set(attributes["k8s.event.action"], body["object"]["action"])
24          - set(attributes["k8s.event.start_time"], body["object"]["metadata"]["creationTimestamp"])
25          - set(attributes["k8s.event.uid"], body["object"]["metadata"]["uid"])
26          - set(attributes["k8s.namespace.name"], body["object"]["involvedObject"]["namespace"])
27
28          - set(log.cache["Normal"], 9)
29          - set(log.cache["Warning"], 13)
30          - set(log.severity_text, body["object"]["type"])
31          - set(log.severity_number, log.cache[log.severity_text])
32          - delete_key(log.cache, "normal")
33          - delete_key(log.cache, "warning")
34
35          - set(attributes["k8s.event.count"], body["object"]["count"])
36
37          - set(body, body["object"]["message"])
```

 Open [receiver/k8seventsreceiver] Labels and attributes extraction from event metadata #39555

done for K8s Node's labels/annotations etc.

 gmlexx on Apr 25, 2025

[@ChrsMark](#) is the absence of these fields in SemConv a real bl

 ChrsMark on Apr 25, 2025

[@ChrsMark](#) is the absence of these fields in SemConv a real

Usually we request covering the SemConv part prior to introduc components.

This is to ensure alignment in definitions and avoid future break

 [odubajDT](#) mentioned this on Apr 23, 2025

 [receiver/k8seventsreceiver] Labels and attributes extraction from event metadata #39555

[odubajDT](#) on /  Closed [receiver/k8seventsreceiver] add extraction rules for labels and annotations #39495

gmlexx wants to merge 1 commit into [open-telemetry:main](#) from [gmlexx:k8s-events-recei...](#)

```
94 + lr := rl.ScopeLogs().At(0)
95 + attrs := lr.LogRecords().At(0).Attributes()
96 +
97 + attr, ok := attrs.Get("k8s.event.labels.testlabel")
```

 [ChrsMark](#) on Apr 22, 2025 Member

We will need to discuss and decide if `k8s.event.label/annotation` is the proper place for this. If so, I guess we would need to define those as Semantic Conventions.

/cc [@TylerHelmuth](#) [@dmityrax](#)

 [TylerHelmuth](#) on Apr 29, 2025 Member

We were planning to deprecate this receiver in favor of the [k8sobjectsreceiver](#) (#24242). If the semconv is willing to define semantic conventions for k8s events we could consider reversing that decision, but my understanding if that we are no longer in the business of defining semconv for log attributes and are instead supposed to follow Otel's Event spec.

 [ChrsMark](#) on Apr 30, 2025 Member

I think focusing on [#24242](#) first would be ideal. It would not worth it adding stuff for k8sevents receiver if it's to get deprecated.

 Closed [receiver/k8sever

 `event.name`

 [therealpandey](#) on Dec 17, 2024 · edited by therealpandey

Very interested in seeing this come to a resolution! Any tim

 [TylerHelmuth](#) on Jan 7, 2025

Not strict timelines. It will probably depend on the Entities :



Let's do something about this

ChrsMark on May 23, 2025 Member

@TylerHelmuth I'm taking a look into this.

My main question here is if we expect that the `k8sobjects` receiver will be a drop-in replacement for `k8sevents` receiver. That doesn't seem possible, right?

At the moment the `k8sobjects` receiver stores the raw event object in the `Body` of the `LogRecord`. Based on <https://github.com/open-telemetry/semantic-conventions/blob/main/docs/general/events.md> and [open-telemetry/semantic-conventions#1651 \(comment\)](#) this is what SemConv and the Specification instruct.

However the `k8sevents` receiver is storing the actual Event message in the `Body` at

```
opentelemetry-collector-contrib/receiver/k8seventsreceiver/k8s_event_to_logdata.go
Line 55 in fac52a
55      lr.Body().SetStr(ev.Message)
```

From my perspective this is a major difference. Because of what the Specification and SemConv instruct `k8sevents` receiver seem to be an invalid implementation right now? If that's the case we need to already start deprecating the `k8sevents` receiver (adding warning messages in README and logs on startup) to set clear expectations and avoid misunderstandings as in [#24242 \(comment\)](#).

Then we can resume the discussion about handling the additional attributes as part of the `k8sobjects` receiver knowing that the outcome will be different from what `k8sevents` used to produce in any case.

Deprecate?





ChrisMark on May 27, 2025

Member

I still need to verify with SemConv SIG what is the most recent guidance for Events, and how this would apply on K8s Events, but I believe the deprecation of the `k8sevents` receiver can start already since the `k8sobjects` receiver can already cover along with the usage of `transform` processor. I would suggest the following plan:

1. Deprecate the `k8sevents` receiver with detailed guidance (in its docs) on how to replace it in honour of the `k8sobjects` receiver and `transform` processor.
2. (optional) Update the `k8sobjects` receiver docs with guidance on how it can be used as a replacement of `k8sevents` receiver.
3. File an issue to make `k8sobjects` receiver, Spec and SemConv compliant. (We could consider adding an opinionated implementation for events probably through a specific setting, initially disabled by-default and in the future enabled by default. We will need to validate with SemConv what schema and handling this will be applying on the raw event object)

[@TylerHelmuth](#) [@dmitryax](#) let me know what you think.



Can SemConv SIG help?

SemConv guidance: document how to define events #1855

Open

 Imolkova opened on Jan 31, 2025 · edited by Imolkova Edits Member ...

See <https://github.com/open-telemetry/semantic-conventions/blob/main/docs/general/how-to-define-semantic-conventions.md#defining-events>

Some areas to cover:

- span vs event (see also [SemConv guidance: document how to define spans #1853](#))
- span attribute vs event
- when to define a new event vs reuse the same event definition across multiple operations
- when to use event attributes vs body ([Logs/Events guidance on what to put to attributes vs body #1651](#))
- when to use complex vs flat attributes ([Semantic Conventions should provide support for code generation of AnyValue \(event/complex types\) fields #1748](#))
- when to set severity and how to do it

Create sub-issue 

Can SemConv SIG help?



Can SemConv K8s SIG help?

Document how K8s Events should be handled #2318

Open



ChrsMark opened on Jun 3, 2025 · edited by ChrsMark

Edits Member ...

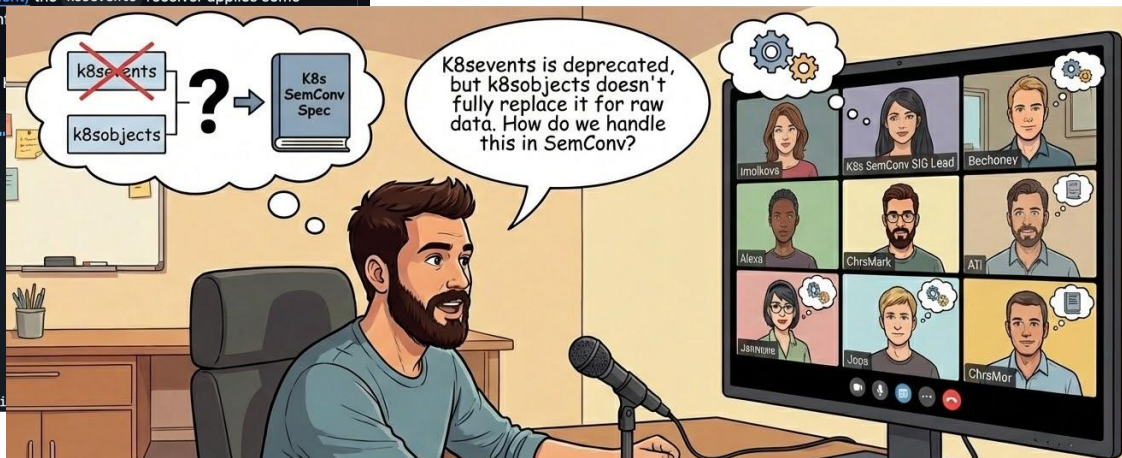
As mentioned at [#1855 \(comment\)](#) we'd like to have documented (or at least have this discussed) how we are going to cover K8s Events.

So far, K8s Events have been handled by the `k8sevents` receiver, however this component was proposed for deprecation being replaced by the more generic `k8sobjects` receiver: [open-telemetry/opentelemetry-collector-contrib#24242](#)

As explained at [open-telemetry/opentelemetry-collector-contrib#24242 \(comment\)](#) the `k8sevents` receiver applies some special handling on the K8s Events it collects from the K8s API. These extract in timestamps, severity etc. The Event's message is also put in the `Log.body`.

On the other hand, `k8sobjects` receiver collects the raw Event object from the

```
2025-05-07T11:45:54.441Z    info    Logs    {"resource logs": 1, "
2025-05-07T11:45:54.441Z    info    ResourceLog #0
Resource SchemaURL:
Resource attributes:
  -> k8s.namespace.name: Str(kof)
  -> k8s.cluster.name: Str(mothership)
ScopeLogs #0
ScopeLogs SchemaURL:
InstrumentationScope
LogRecord #0
ObservedTimestamp: 2025-05-07 11:45:54.245915752 +0000 UTC
Timestamp: 1970-01-01 00:00:00 +0000 UTC
SeverityText:
SeverityNumber: Unspecified(0)
Body: Map({"object":{"apiVersion":"v1","count":15,"eventTime":null,"fi
```



KubeCon



CloudNativeCon

Europe 2026

Progress!

The screenshot shows a GitHub discussion thread. On the left, a comment by **dmitryax** on Jun 16, 2025, discusses maintaining Kubernetes signals and mentions a plan to make the code re-usable. A reply by **ChrsMark** on Jun 19, 2025, mentions two issues created: [Make k8sevents receiver SemConv compliant #40823](#) and [Make code re-usable between k8sevents receiver and k8subjects receiver #40825](#). Below this, another comment by **dmitryax** on Jun 18, 2025, states "Closing as 'won't do'". The thread includes reaction icons (thumbs up, thumbs down) and a "Member" badge.

dmitryax on Jun 16, 2025

The issue was originally s
be maintaining Kubernetes
signals, I believe it's impc
signals.

While we could keep only
think that would provide :
processor to other receiv

Additionally, we're planni
k8clusterreceiver). This

Given this, I'l
+ entities. Th

Instead of de
more code. f
on top.

ChrsMark on Jun 19, 2025

Two follow-up issues created:

- [Make k8sevents receiver SemConv compliant #40823](#)
- [Make code re-usable between k8sevents receiver and k8subjects receiver #40825](#)

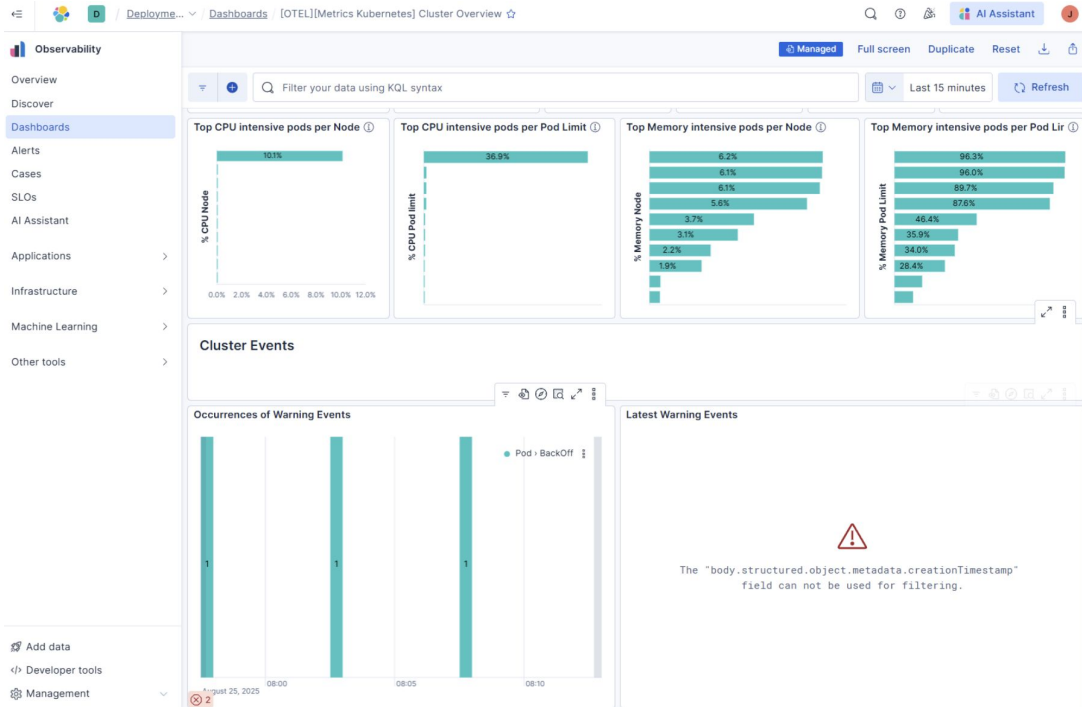
dmitryax on Jun 18, 2025

Closing as "won't do"

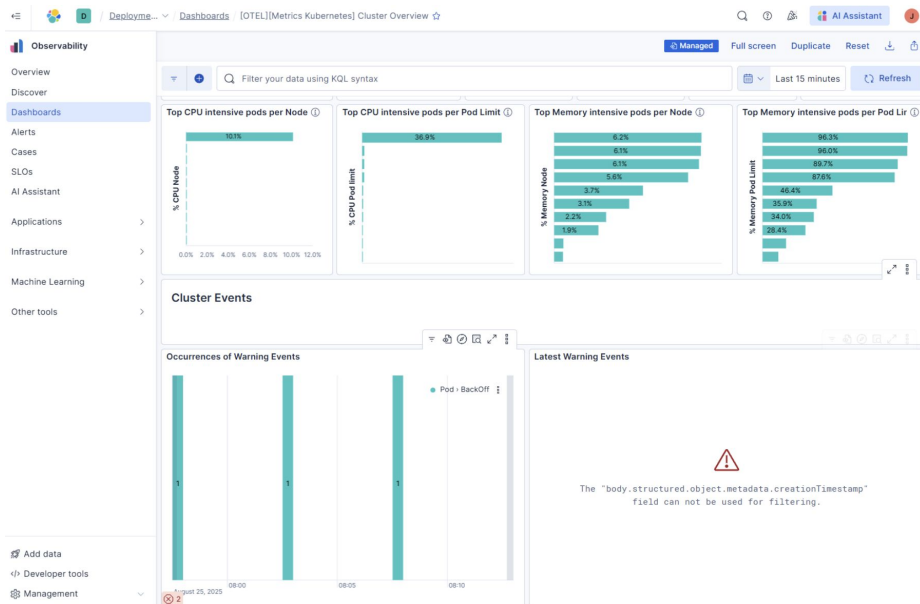
s SemConv guidance mentioned above.

eivers, the former one should be
tion (not sure if [#39555](#) falls into that

Few months later...the “real” customer(s) issue



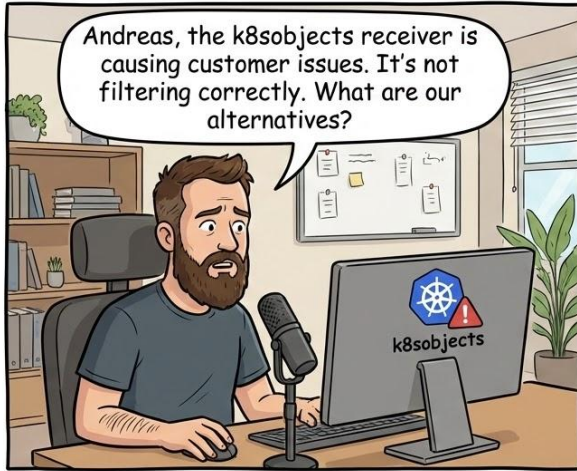
Few months later...the “real” customer(s) issue



Multiple customers
started reporting that issue



Evaluate alternatives



We aim for minimal
custom configurations



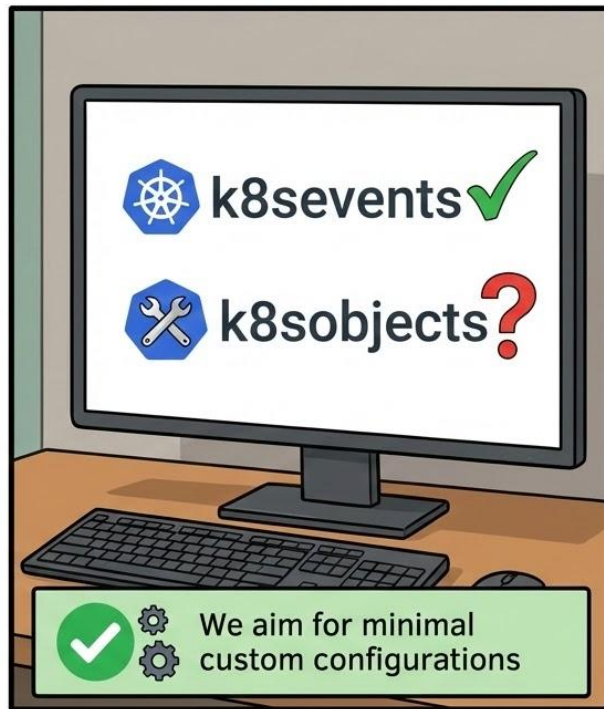
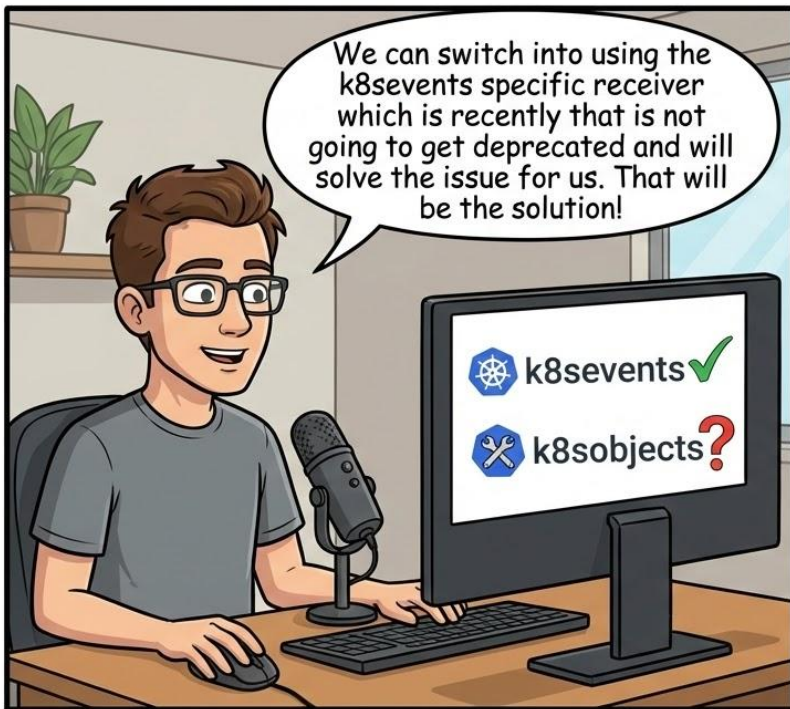
KubeCon



CloudNativeCon

Europe 2026

We can use k8sevents receiver



How it was solved confidently

Evaluate the usage of k8sevents instead of k8subjects

Include k8sevents in EDOT

Switch helm charts' default configs to use k8sevents

Tune the dashboards

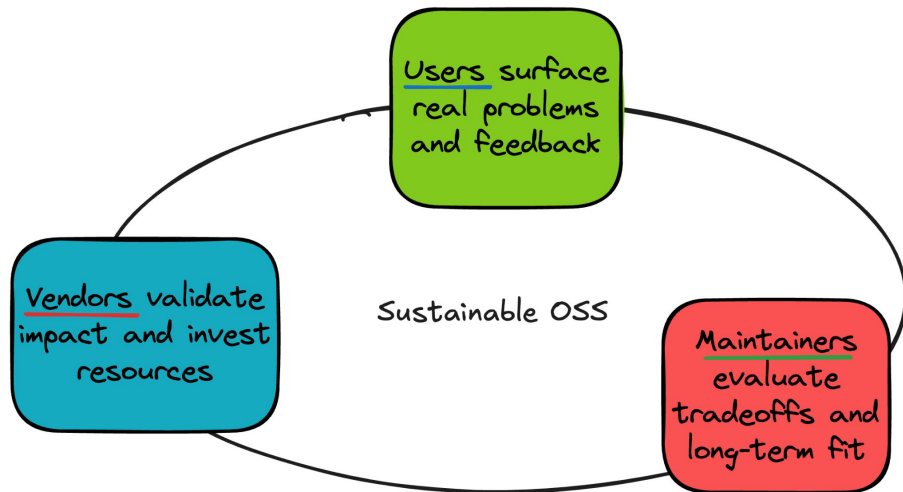
Deprecate k8subjects' usage

Outcome

1. Happy customers
2. Support issues resolved
3. Support creates a knowledge article
4. Community and maintainers are happy (already)
5. Everyone is happy eventually!

Outcome

1. Happy customers
2. Support issues resolved
3. Support creates a knowledge article
4. Community and maintainers are happy (already)
5. Everyone is happy eventually!





KubeCon



CloudNativeCon

Europe 2026



Closing thoughts (what I've learned)

1. **Show up and start small**
2. **Align and build trust**
3. **Do the work and be patient**
4. **Step up intentionally**
5. **Be kind, open, and future-focused**
— open source is not a race, it's a marathon (K.H.), so build for the long run.

Thank you!



KubeCon



CloudNativeCon

Europe 2026

Thank you!

