Falco

KubeCon | CloudNativeCon
North America 2026

Falco Never Flies Alone: Runtime Security as an Ecosystem

Alessandro Cannarella, vCluster
Leonardo Grasso, Sysdig

Thursday November 12, 2026 · 2:40pm MST · Salt Palace | Room 251 D-F



About Us



Follow

Alessandro Cannarella

c2ndev

Falco Core Maintainer

vCluster

github.com/c2ndev



Follow

Leonardo Grasso

leogr

Falco Core Maintainer

Sysdig

github.com/leogr

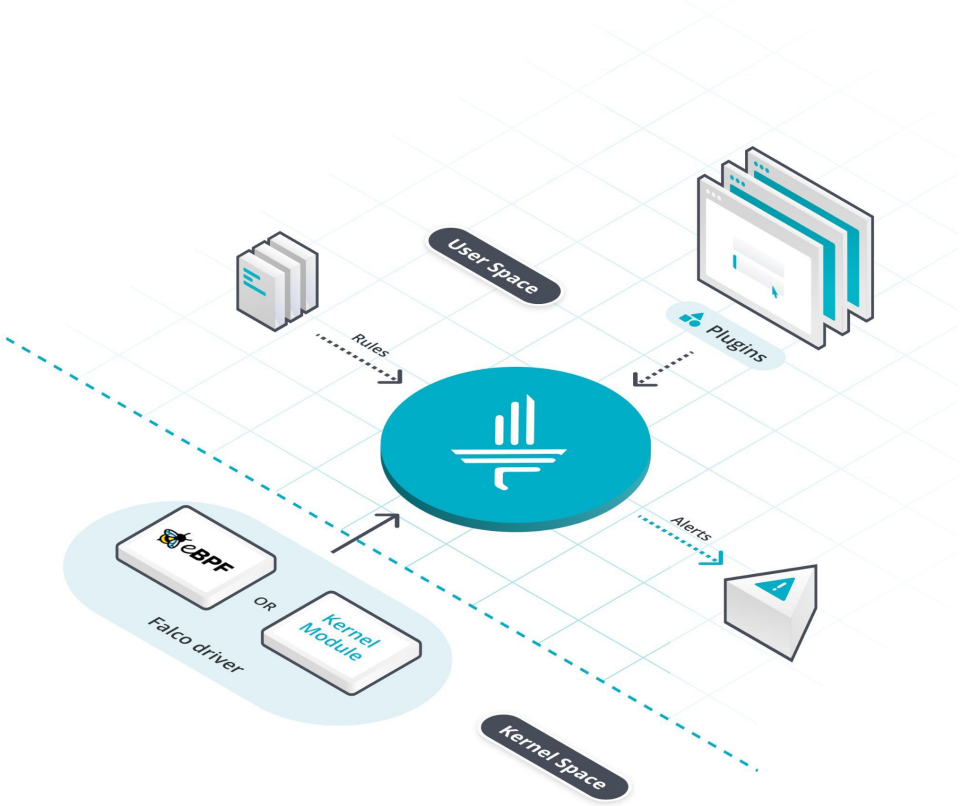
Falco

What's Falco?

Detect security threats in real time

Falco watches your hosts, containers, Kubernetes and cloud. It checks every event against your rules and alerts you the moment something looks wrong.

Like a security camera for your cloud



Falco

Falco

10 years of Falco

A decade of development, adoption and community contributions

2016
First commit

2018
CNCF Sandbox

2020
CNCF Incubation

2024
CNCF Graduation

2026
10 years

Ten years in, Falco has never flown alone.
So what did 2026 bring?

Community announcement ↗

Falco

Strong Wings: Falco in 2026

A stronger core, and what comes next

Leaner Engine, Sharper Rules

Modern capture stack

Legacy eBPF, gVisor and gRPC output removed in 0.44. Faster process lookups.

0.44 ↗ · 0.45 ↗

More expressive rules

New oneof, anyof, allof modifiers. Stricter schemas, raw-byte matching.

0.44 ↗ · 0.45 release ↗

Safer forensic captures

Capture files now have a configurable size limit, on top of the 2025 recording feature.

Capture update ↗

Wider detection coverage

New sandbox rules for suspicious NPM installs and GPU hijacking (off by default).

Rules 6.1.0 ↗ · 6.2.0 ↗

Falco

Smoother to Run, Safer to Operate



Plugins that see more

Kubernetes audit now covers init and ephemeral containers. Container metadata got better timeouts, recovery and reload.

K8s audit 0.18.0 ↗
Container 0.7.6 ↗



Smoother reloads and upgrades

Incubating reload status and control APIs, correct DEB/RPM upgrades, and DKMS installs preserved by Falcoctl.

Falco 0.45 ↗
Falcoctl 0.14.2 ↗



Security and reliability, through 0.45.2

October fixes for metrics races, file-output recovery, reload failures, certificate trust and dependencies.

0.45.2 blog ↗

Falco

New Directions

Prempti

Experimental

Falco meets AI coding agents

Falco policies applied to an agent's tool calls: see what it does, then allow, deny or ask. Works with Claude Code, with Codex support arriving in 0.3.

Launch blog ↗ · v0.3.0-rc1 ↗

Stratoshark

Release candidate

Progress toward 1.0

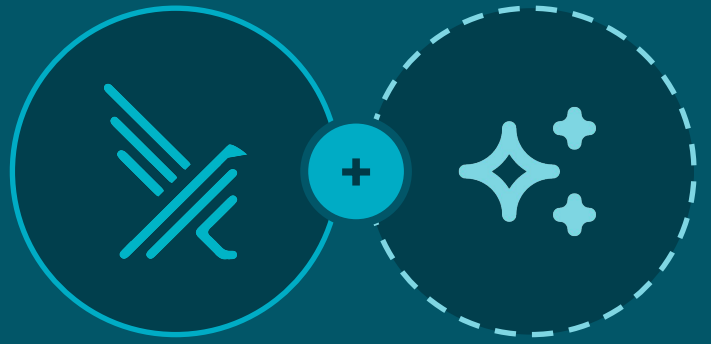
1.0.0rc1 brings the investigation workflow closer to Falco, with Falco libs 0.26.1 and raw capture-block processing.

Release notes ↗

Falco

Falco Has a New Friend: Falco Expert

- Falco knowledge and practical skills for your AI coding agent
- Help understanding the project, developing changes, and preparing contributions
- For your first contribution, and your next one



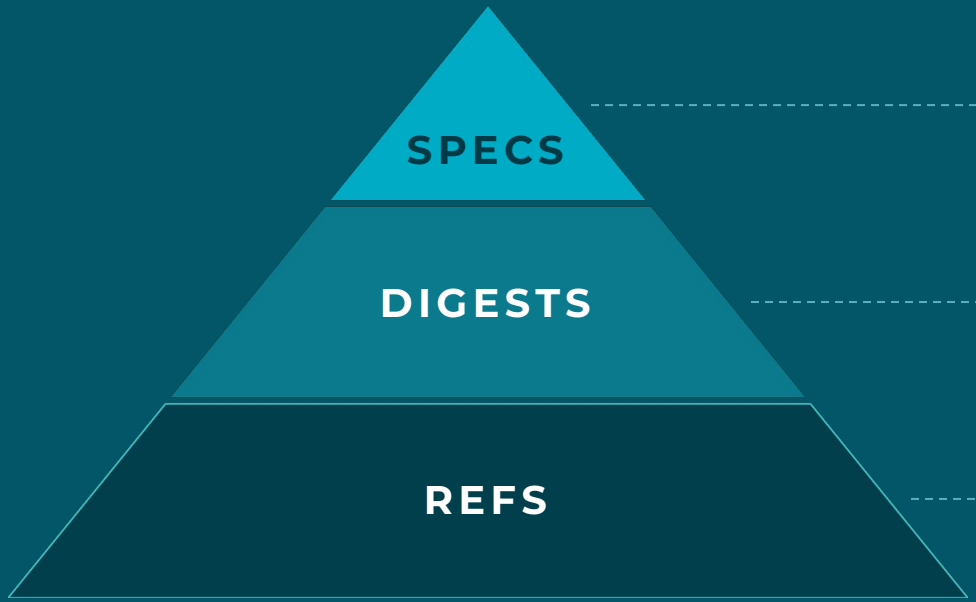
Falco **Your coding agent**

Falco never flies alone, and neither do its contributors.

Falco

A Knowledge Base Your Agent Can Navigate

From source material to structured understanding.



Specs
Technical specifications describing architecture and behavior

Digests
Component summaries with references back to their sources

Refs
Falco code and documentation pinned to specific versions

Knowledge provides context. Skills guide practical work.

Falco

Dig Deeper: Follow the Evidence

How a question becomes a verifiable answer.

1 Break down

the question into focused investigations

→

2 Navigate

specs → digests → original sources

→

3 Verify

check the evidence and distinguish facts from uncertainty

→

4 Explain

return a cited answer and a shareable report

>_ EXAMPLE PROMPT

```
❯ Dig deep into why Falco's modern eBPF driver drops events when a workload
rapidly creates processes. Help me reproduce the issue, trace where events are
lost, and determine whether it needs a configuration change or a code fix.
```

Falco

Practical Skills for Your Next Contribution

Understand → develop → validate → self-review.



`falco-dev`

Build, test, and debug Falco and its core libraries in a development container.



`falco-rules-author`

Write detection rules, validate them, and investigate false positives.



`falco-reviewer`

Self-review your PR for correctness, security, and breaking changes before requesting maintainer review.

Bring a change you understand, with evidence others can review.

Falco

Help Us Make Contributing Easier

Try Falco Expert on your next contribution.

01

Set it up

Ask your AI coding agent to follow the Falco Expert setup guide

02

Bring a real task

Bring an issue or a question, and use the skills to develop and review your contribution

03

Share back

Share feedback, missing knowledge, and workflow improvements

`github.com/leogr/falco-expert ↗`

You don't need to know every corner of Falco to start contributing.

Falco

Falco

Who Else Flies With Falco?

The ecosystem around the engine

One Engine, and a Whole Stack Around It

Run

Falco Operator

Runs Falco and its companions on Kubernetes, and keeps them the way you declared.

Route

Falcosidekick

Takes every Falco event to the tools your team already uses.

React

Falco Talon

Turns a detection into an action inside the cluster.

Falco

Falco

Falco Operator

From tech preview to the way you run Falco on Kubernetes

The Story So Far

v0.1
May 2025

Tech preview. A first taste of running Falco the Kubernetes way.

v0.2
March 2026

Production-ready. Cleaner API, rules from ConfigMaps, the whole stack as Components.

v0.3 + v0.4
May and June 2026

Helm chart as a first-class install, Falco 0.44, a lot of day-2 polish.

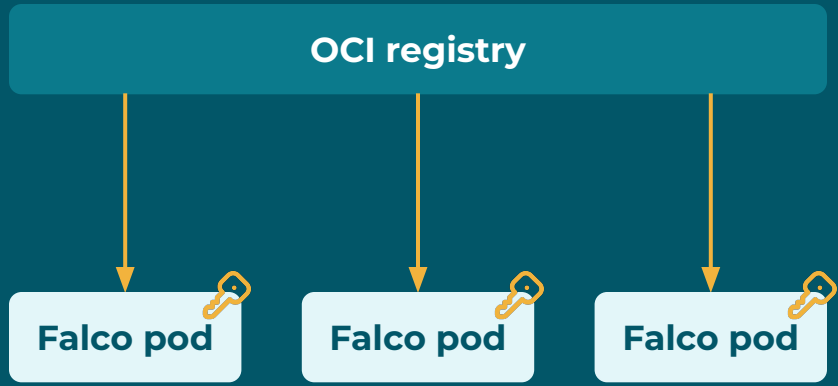
v0.5
November 2026

Four release candidates and 176 commits later: artifact delivery, rebuilt.

Falco

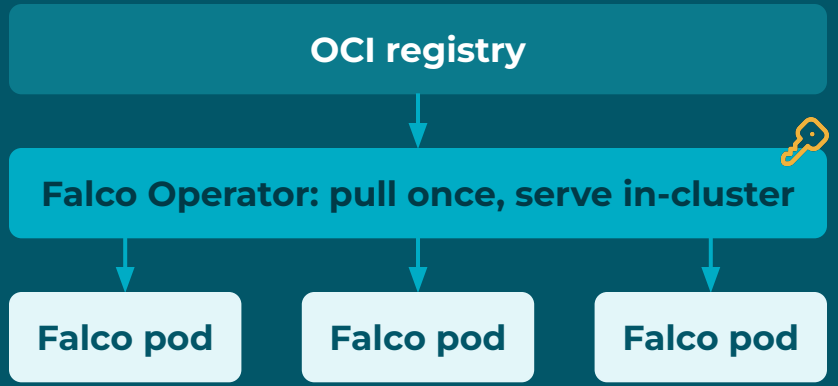
v0.5: What Changed Under the Hood

Until v0.4



Every pod pulls from the registry.
Every pod needs the credentials.

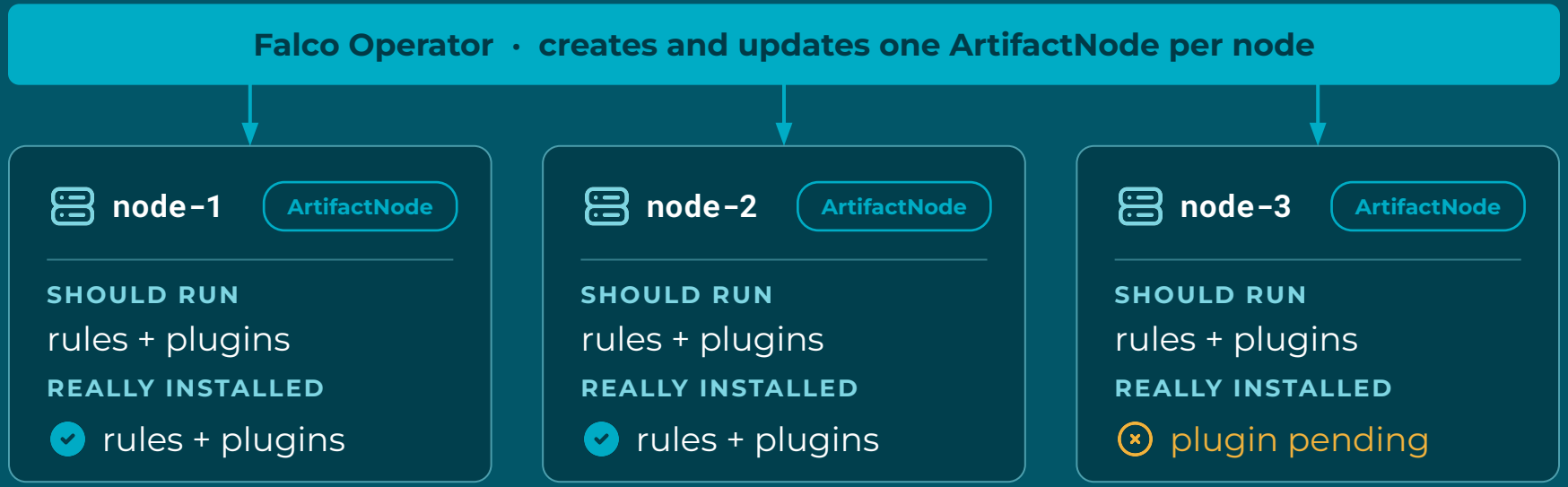
v0.5



One pull, one revision for every node.
Credentials stay with the operator.

Falco

v0.5: ArtifactNode

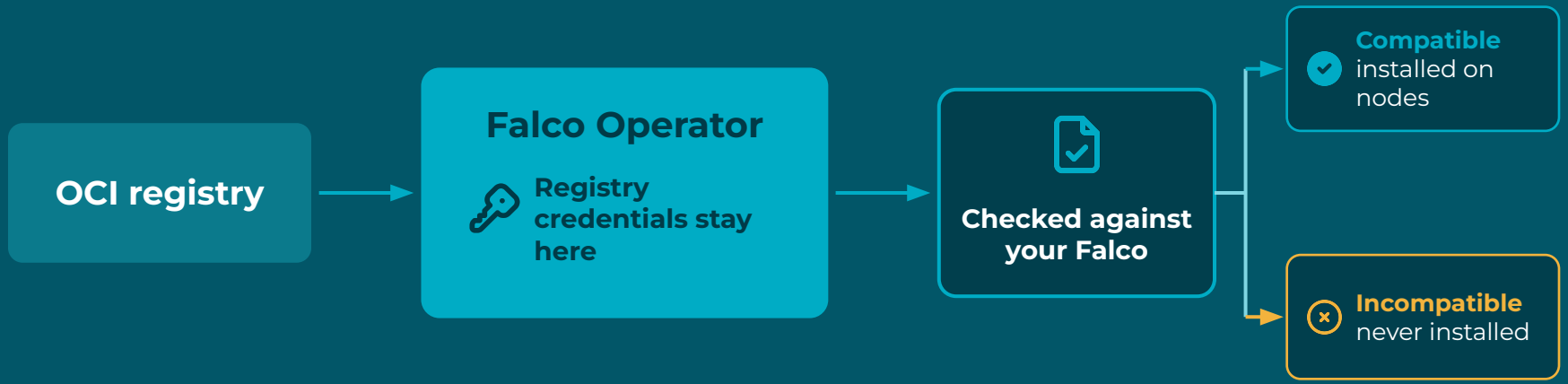


One per node: what it should run, and what it really installed.

You never write it. The operator does.

Falco

v0.5: Safer by Default



Artifacts are checked against your Falco before install.

Registry credentials stay with the operator.

Falco

Falco

One more thing...

Have you ever heard about Falcosidekick?

Falco

Falcosidekick v3

Rewritten from scratch

`3.0.0-beta.1`

What v3 Does Differently



A real pipeline

- A queue per output
- Retry + circuit breaker
- Drains on shutdown



Outputs as drivers

- One contract, one schema
- Validation, docs and UI forms from it



Config you can trust

- Validated at startup
- Live reload
- Edit from the API or UI

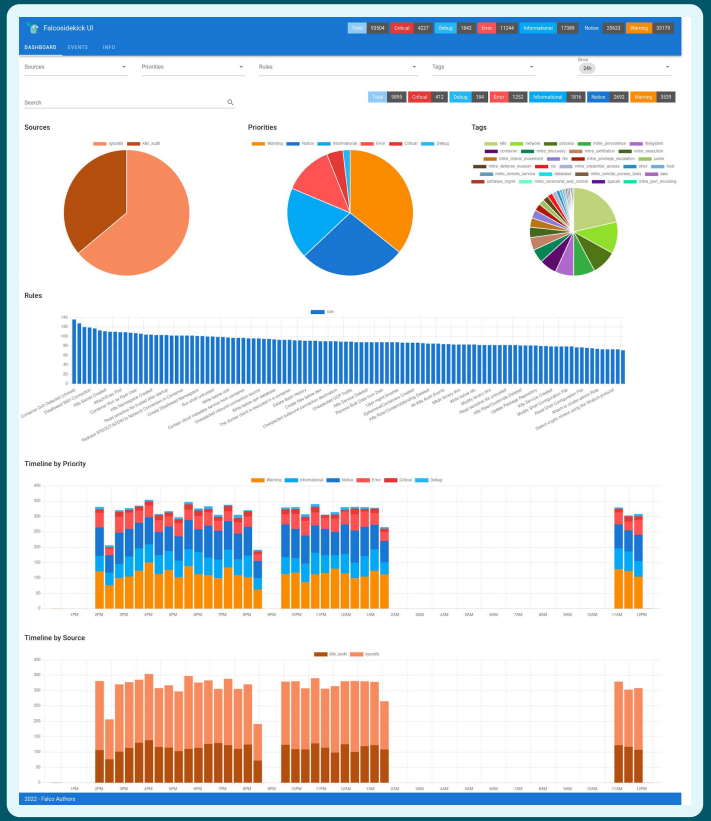


One binary, UI included

- Dashboard built in
- No Redis
- No second service

Falco

The UI, Before

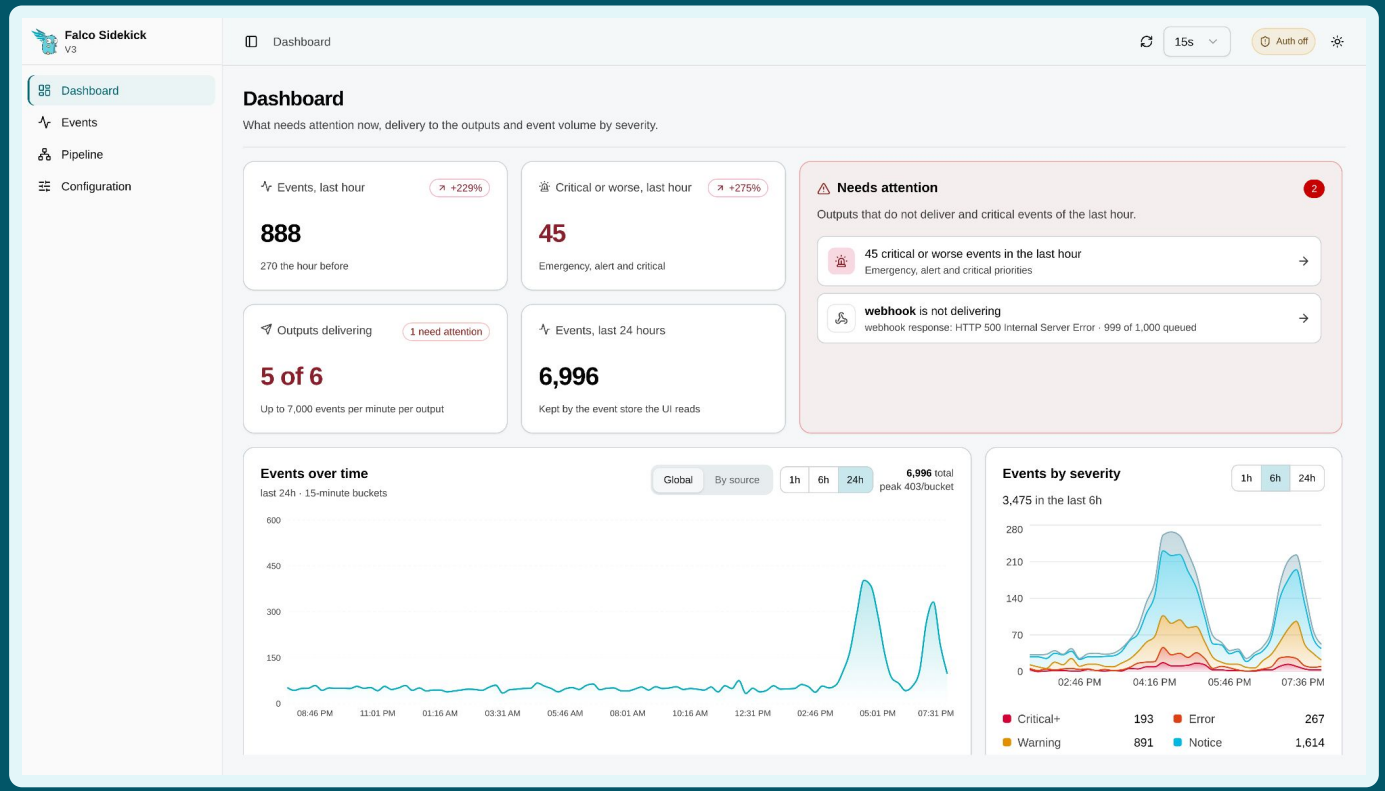


It worked.

It also needed its own Redis.

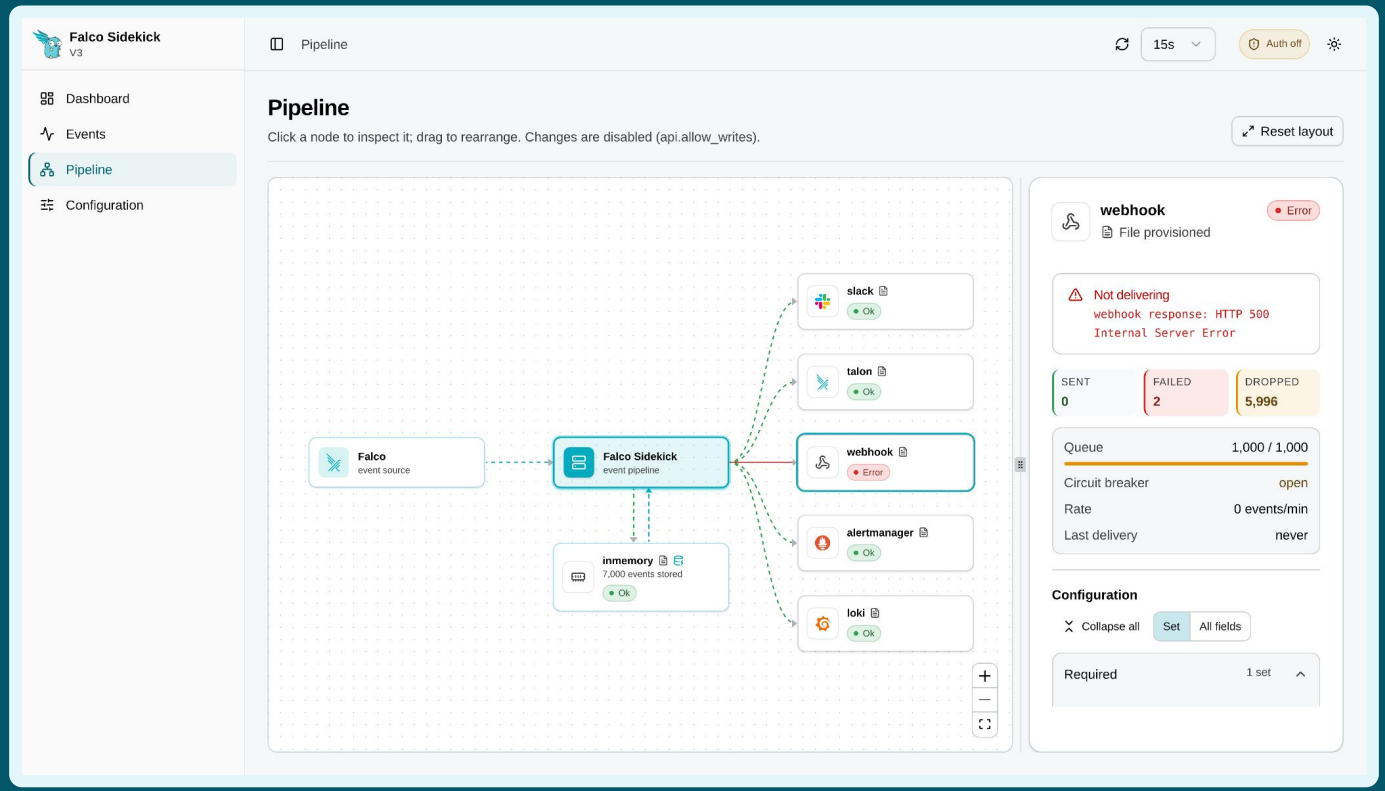
Falco

The UI, Now



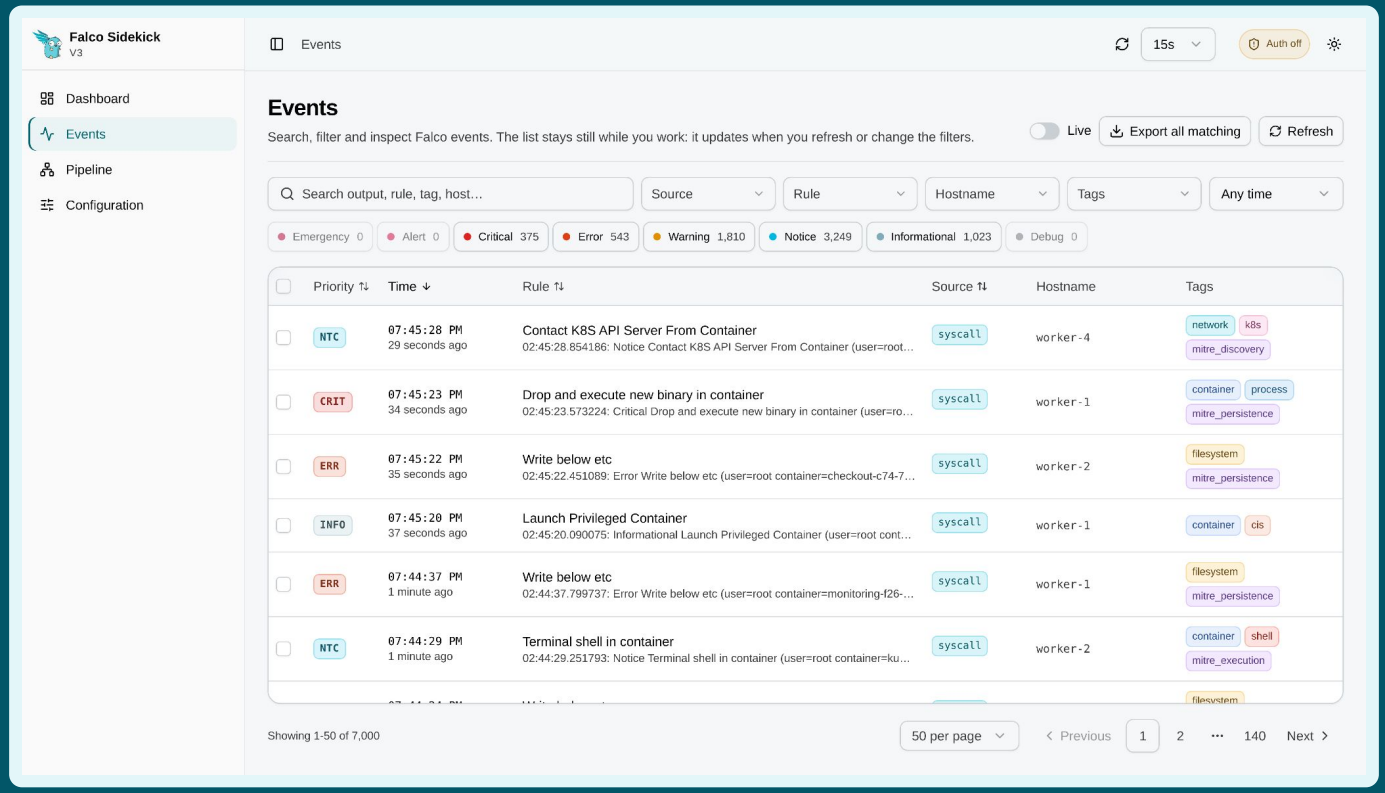
Falco

It Tells You What Is Broken



Falco

Monitor, Search, Filter and Export



Falco

What's in Beta 1

A first taste of v3. Short list, big change.



24 outputs, each with a docs page

`falcosidekick`

API | Web UI

1 binary, with the API and web UI inside

live reload

memory | SQLite | Postgres

0 restarts to add or change an output

3 places to keep the outputs you create

Falco

24 Outputs in Beta 1


Slack


Microsoft Teams


Discord


Alertmanager


PagerDuty


Opsgenie


Datadog


Grafana OnCall


Loki


Elasticsearch


Syslog


OpenTelemetry


Apache Kafka


Google Pub/Sub


Amazon SNS


Amazon SQS


Amazon S3


Redis


In-memory store


Webhook


CloudEvents


Email (SMTP)


Falco Talon


Policy Report

Falco

Try It, Break It, Tell Us

Image

```
falcosecurity/falcosidekick:3.0.0-beta.1
```

Helm

```
$ helm install falcosidekick falcosecurity/falcosidekick --devel
```

v2 is not going away: it lives on the 2.x branch for security and critical fixes

It is a beta. Break it, then tell us in **#falco** or in a **GitHub issue**

Falco

Join Our Community and Reach Out to Us

#falco

channel on **kubernetes.slack.com**

Community Calls

every 2 weeks on Wednesday

Mailing list

cncf-falco-dev@lists.cncf.io

falco.org

Docs, blog and everything Falco

Falco

Falco

Thank you!

falco.org

github.com/falcosecurity

#falco on Kubernetes Slack