

From 0 to `</>` my one year serendipity journey with OpenTelemetry

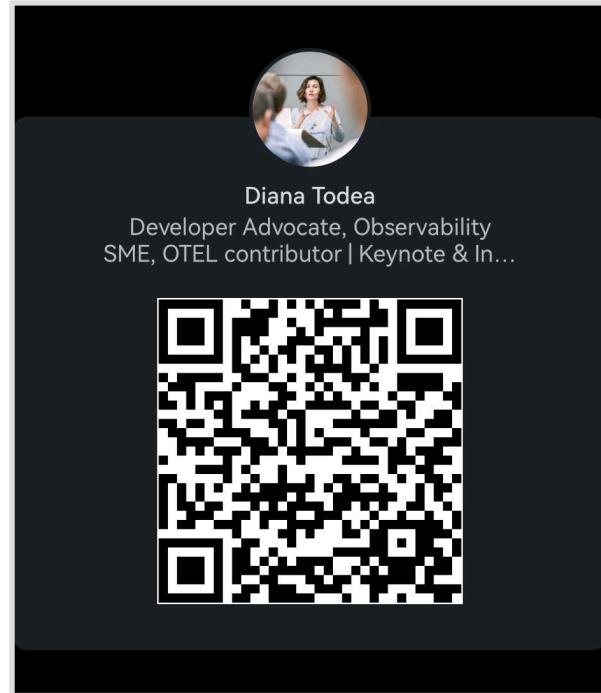


19th of June 2025

Diana Todea
Technical Advocate @Aircall

[github](#)

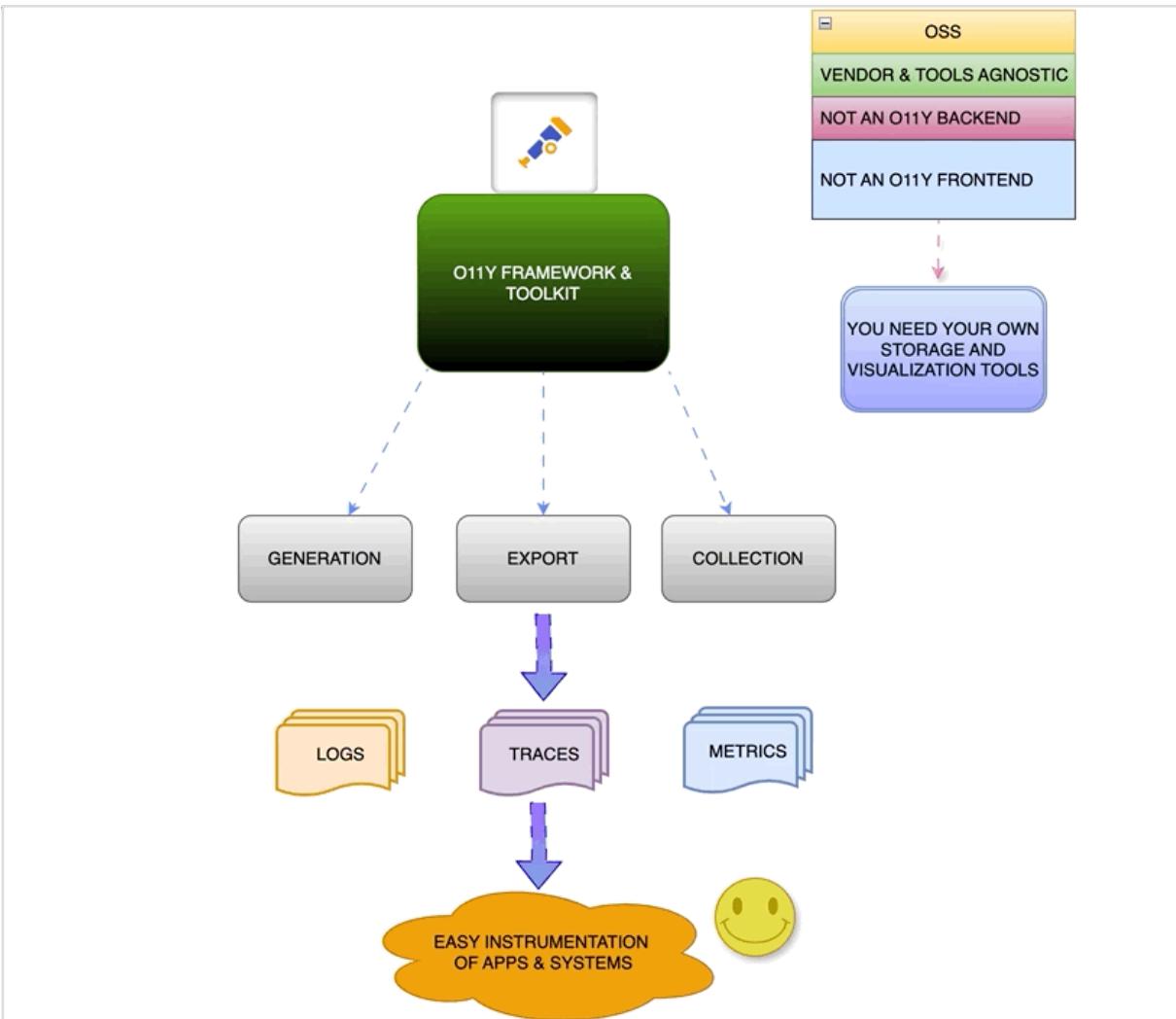
@didiviking/Conferences_Talks



a beginning



“OTel craze and an SRE’s curiosity”



“The OTel call for olly SMEs”

The screenshot shows the CERTIVERSE platform interface. At the top, there's a dark header with the CERTIVERSE logo, a star icon indicating 1530 items, a bell icon, and a 'CONTACT US' button. Below the header, a dropdown menu shows 'Organization: Cloud Native Computing Foundation'. The main area has tabs for 'Item Writing' (1) and 'My Items' (21). A search bar allows filtering by 'Organization' and 'Name'. A table lists a single item: 'Cloud Native Computing Foundation' under 'Organization' and 'OpenTelemetry Certified Associate (OTCA)' under 'Name'. A 'WRITE ITEMS' button is at the bottom right of the table.



The promotional page for the OpenTelemetry Certified Associate (OTCA) certification. At the top, a 'CERTIFICATION' badge is shown above the title 'OpenTelemetry Certified Associate (OTCA)'. To the right is a circular badge for 'CLOUD NATIVE COMPUTING FOUNDATION CNCF'. The main title is 'OpenTelemetry Certified Associate (OTCA)'. Below the title, a subtext reads: 'As cloud native systems grow more complex, the demand for professionals who can leverage telemetry data is growing rapidly. Open new career paths – prove your expertise in OpenTelemetry – the industry standard for tracing, metrics & logs.' A section titled 'OTCA includes:' lists two items: '12-months to schedule & take the exam' and 'Two exam attempts'. At the bottom, a note says 'Now Available! You can bundle OTCA with an annual THRIVE subscription to get access to more than 100 educational products and skillCreds for only \$495!' To the right, there's a graphic of a certificate badge with 'OpenTelemetry CERTIFIED Associate' text.

a middle



“Gee, what a great start, can I do more?”



“An SRE’s purpose is to turn
everything into OTel”

“How about a POC?”

I played with OTEL and I liked it

 Diana Todea
4 min read · Jan 7, 2025

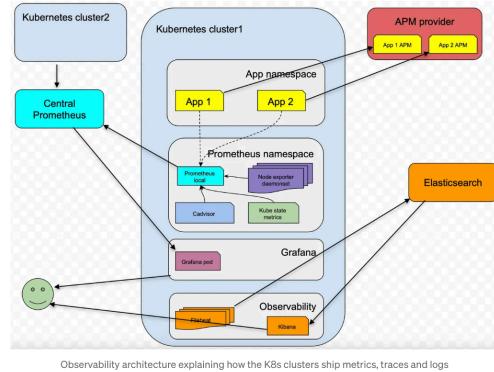
25 1

...

As mentioned in my previous [article](#), in 2024 I had the opportunity to work on an observability project where I wrote a proof of concept on [OpenTelemetry](#). As an SRE I saw the window to experiment with one of the most popular CNCF projects on distributed tracing and provide an open source alternative to monitor the performance of my client's applications. I had the chance to give a more in depth talk on this topic at KCD Accra Ghana as well as at [OSMC 2024](#). You can access my presentations directly on my [Github repo](#).

Architecture

This particular observability architecture was mainly composed of [Prometheus metrics](#), [Victoria Metrics](#), [Grafana](#) for visualizations and alerting, [ELK](#) for logging and a commercial APM vendor. A fragmented observability architecture leads to a confusing developer and user experience, working in observability for almost 5 years now it's easy to say that most people prefer having a unified monitoring pane of glass, which is easier to onboard and manage.



Take-aways:

- great SRE+SWE collaboration
- support the common format and protocol for collecting telemetry data
- vendor neutrality

We loved it, but:

OTel considers profiling and instrumenting as a different scope

- instrumentation comes with a lot of manual effort
- not the smoothest onboarding due confusing docs
- we didn't account for not enough engineering time/manpower
- we weren't satisfied with OSS visualization frontend
- OTel is not yet a stable project: what type of support we will get in production?

Provide your feedback to the OTel community

<https://github.com/open-telemetry/opentelemetry-specification/issues/4255>

The screenshot shows a GitHub issue page for issue #4255. The title is "Truly auto instrumentation #4255". The issue is labeled "Open". A comment by user "ostrolucky" is displayed, dated Oct 11, 2024. The comment discusses the limitations of the current opentelemetry-auto-* packages and proposes a solution where OpenTelemetry can automatically find unexpected bottlenecks. It also suggests renaming the opentelemetry-auto concept to opentelemetry-preset.

What are you trying to achieve?
Having a way to find unexpected bottlenecks
Additional context.
My biggest problem with Opentelemetry is that the way it seem to work is you have to specify which functions to instrument. There are opentelemetry-auto-* packages, however all those are is predefined lists of such functions. No matter if you use opentelemetry-auto-* packages or not, problem remains: You have to specify exact functions to instrument, which is not how commercial solutions do it.
Why is this a problem?
Since you have to pick functions to instrument, once bottleneck happens in a function you didn't expect would be a bottleneck, you will have no clue from your spans where is the bottleneck, since you don't instrument it. All you are left with at that point is to check all function calls from the place that is identified by parent span, add some or all of them to your instrumentation and redeploy.
To make opentelemetry more useful, we should have a way to instrument on something else than some static list of functions, for example random sampling instrumentation of all function calls, or any function call which exceed time treshold. That would be closer to the way people are used to and I would argue more useful than what we have now, which leaves you quite stuck once you run into some unexpected slowdown and struggling to find the root cause.
I would also suggest to rename current opentelemetry-auto concept to opentelemetry-preset, as auto here is very misleading if what those packages mostly do is to predefine list of functions to trace

[#otel-ebpf-instrumentation](https://github.com/open-telemetry/community/issues/2406)

The screenshot shows a GitHub issue page for issue #2406. The title is "[Donation Proposal]: Belya, eBPF auto-instrumentation tool for metrics and traces #2406". The issue is labeled "Open". A comment by user "grcevski" is displayed, dated Oct 23, 2024. The comment proposes a donation of Belya, an eBPF-based auto-instrumentation tool. It describes Belya's features and its use cases. The right side of the screen shows the GitHub sidebar with various project details.

[Donation Proposal]: Belya, eBPF auto-instrumentation tool for metrics and traces #2406
Open
grcevski opened on Oct 23, 2024 - edited by grcevski
Description
Grafana Labs would like to offer the donation of Belya to the OpenTelemetry project.
Belya is a mature eBPF-based auto-instrumentation tool for OpenTelemetry metrics and traces, for multiple languages and protocols. It enables cluster-wide/system-wide auto-instrumentation of applications without the need for application code/configuration changes or application restarts. To achieve this, Belya uses a combination of protocol-level instrumentation based on network events and language/runtime-level instrumentation where needed. While Belya works on bare metal installations, virtual machines, etc., the tool is also fully Kubernetes-aware and can be deployed as a daemonset or as a sidecar. Belya is used by a number of customers in production, including Grafana Labs itself for the Grafana Cloud hosted offering.
Some of the main uses of Belya are:

- Provide auto-instrumentation for programming languages where OpenTelemetry SDK zero-code auto-instrumentation is not supported, such as Rust, C++, Erlang, Zig, Ruby, Swift, Perl, Lua, Dart, R, Java GraalVM Native, Julia...
- Provide auto-instrumentation for legacy applications, where it's not easy to migrate the codebase to the OpenTelemetry SDK compatible frameworks.
- Provide auto-instrumentation for applications where the source is not available or are proprietary and/or distributed in binary form.
- Provide a unified way to capture application-level metrics across all different technologies used by a customer.
- Provide network-level metrics, regardless of the L3/L4/L7 protocol used for the purpose of building service graphs and reachability reports.
- Provide process-level metrics for instrumented applications.

Some of the core features of Belya include:

“Don’t despair if it’s not in production.”



A screenshot of a GitHub pull request comment thread. Diana Todea (@krol) at 2:41 PM asks about adding heading IDs to documentation. @krol replies that it's the same issue as their previous PR (#6134). The thread shows two more comments from @krol (#6371) and a reply from @didiViking. The last reply was 2 months ago.

Diana Todea 2:41 PM
Hola, por ese PR no entiendo muy bien el ultimo comentario para añadir los heading IDs, es la misma cosa que falla por mi ultimo PR. Si sabes de que va o como se soluciona, podemos hacer una llamada. MGTV

#6134 feat: [es] Create docs/contributing/style-guide.md
open-telemetry/opentelemetry.io | Jan 31st | Added by GitHub

#6371 feat: [es] Create docs/contributing/pr-checks.md
open-telemetry/opentelemetry.io | Feb 18th | Added by GitHub

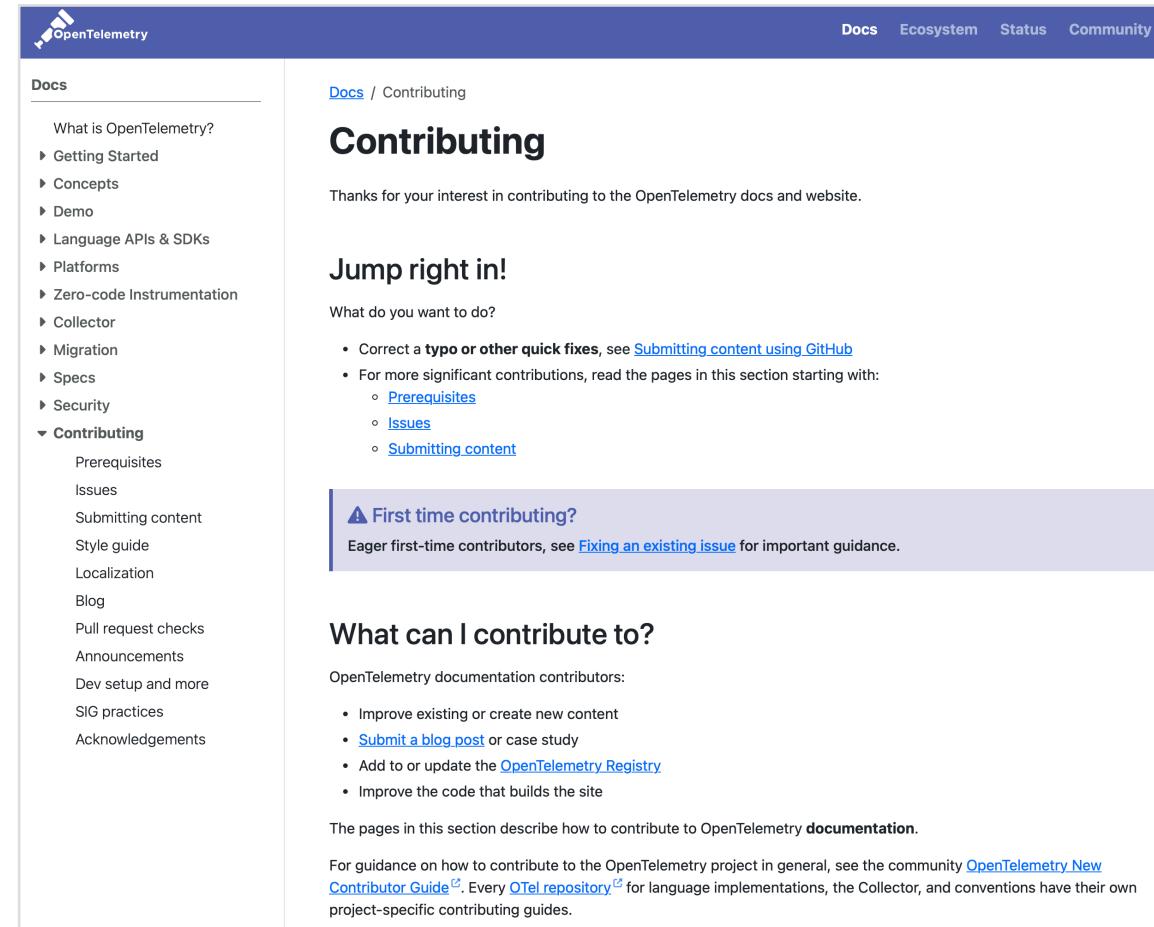
1 reply 3 replies Last reply 2 months ago

Don't minimize no code contributions

A screenshot of three GitHub pull requests from the open-telemetry/opentelemetry.io repository:

- #6371 by didiViking was merged yesterday • Approved. A comment from lang:es is visible.
- #6134 by didiViking was merged on Feb 19 • Approved. A comment from lang:es is visible.
- #6088 by didiViking was merged on Feb 14 • Approved. A comment from lang:es is visible.

Pay it forward



The screenshot shows the OpenTelemetry documentation website. At the top right, there are links for Docs, Ecosystem, Status, and Community. The main navigation bar has a "Docs" section with a dropdown menu containing "What is OpenTelemetry?", "Getting Started", "Concepts", "Demo", "Language APIs & SDKs", "Platforms", "Zero-code Instrumentation", "Collector", "Migration", "Specs", "Security", and "Contributing". The "Contributing" item is expanded, showing sub-sections like Prerequisites, Issues, Submitting content, Style guide, Localization, Blog, Pull request checks, Announcements, Dev setup and more, SIG practices, and Acknowledgements. On the right side of the page, the URL "Docs / Contributing" is shown above the title "Contributing". Below the title, a message says "Thanks for your interest in contributing to the OpenTelemetry docs and website." A large heading "Jump right in!" is followed by the subtext "What do you want to do?". A bulleted list provides guidance: "Correct a typo or other quick fixes, see [Submitting content using GitHub](#)", "For more significant contributions, read the pages in this section starting with: [Prerequisites](#), [Issues](#), and [Submitting content](#)". A callout box titled "⚠ First time contributing?" contains the text "Eager first-time contributors, see [Fixing an existing issue](#) for important guidance." Below this, a section titled "What can I contribute to?" lists "OpenTelemetry documentation contributors:" with a bulleted list: "Improve existing or create new content", "Submit a blog post or case study", "Add to or update the [OpenTelemetry Registry](#)", and "Improve the code that builds the site". A note states that the pages in this section describe how to contribute to OpenTelemetry documentation. Finally, a note about the community states: "For guidance on how to contribute to the OpenTelemetry project in general, see the community [OpenTelemetry New Contributor Guide](#). Every [OTel repository](#) for language implementations, the Collector, and conventions have their own project-specific contributing guides."

- Nurture local communities!
- Spanish speakers needed!



#otel-docs-localization
#otel-localization-es

<https://community.cncf.io/cloud-native-valencia/>



Share with communities

OSMC SUBMIT A TALK

FROM ZERO TO DEVELOPER: MY ONE YEAR SERENDIPITY JOURNEY WITH OPentelemetry

Becoming a contributor to an open-source project is a transformative step in any developer's career. This session explores the journey from first-time contributor to active developer, covering best practices for navigating project communities, understanding codebases, and making meaningful contributions. Learn strategies for selecting the right project, mastering collaboration tools, and embracing the culture of open-source development. The audience will be inspired about my one year journey with the open source project OpenTelemetry, how I have built a proof of concept for it and achieved developer status for this project. By the end of this talk, the public will gain insights into the tools to become a better developer and how to build more engagement with the community.

SPEAKER

Diana Todea
Aircall

DIANA TODEA Developer Advocate at Aircall

Diana is a Developer Advocate at Aircall. She has worked as a Senior Site Reliability Engineer focused on Observability. She is passionate about serverless, SecOps and machine learning. She is an active contributor to the OpenTelemetry open source project and supports women in tech.

Sessions

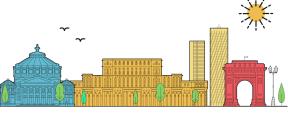
From zero to developer: my one year serendipity journey with OpenTelemetry

Go back

5-6 MAY 2025

Cloud Native Days Romania

The Kubernetes & Cloud Native community will gather at the Radisson Blu Bucharest, Romania. Join us for one day technical event loaded with exciting talks and networking opportunities. Cloud Native Days Romania is aimed at developers, platform people, and other IT professionals with an interest in cloud native technologies.



From zero to developer: my one year serendipity journey with OpenTelemetry ⭐

2025-06-19 15:05 - 15:30, Grote Zaal

Becoming a contributor to an open-source project is a transformative step in any developer's career. This session explores the journey from first-time contributor to active developer, covering best practices for navigating project communities, understanding codebases, and making meaningful contributions. Learn strategies for selecting the right project, mastering collaboration tools, and embracing the culture of open-source development. The audience will be inspired about my one year journey with the open source project OpenTelemetry and how I have built a proof of concept for it and achieved developer status for this project. By the end of this talk, the public will gain insights into the tools to become a better developer and how to build more engagement with the community.

open ObservabilityCon + Tel Community Day

June 26 Denver, Colorado

#OpenOttCon #OTelDay

From zero to developer: my one year serendipity journey with OpenTelemetry

Cloud Native Santo Domingo

Jul 12, 4:00 - 5:30 PM (GMT+2)

Virtual event

Why feedback matters for open source



Diana Todea · 3 min read · Jan 7, 2025

My 2024 has been a year where I finally immersed myself in the open source realm. It has been on my mind for a while to start contributing to an open source project, however I wanted to pick the right project that resonated with my current on-going work.

Putting doubts aside, I was really impressed with [OpenTelemetry](#)'s popularity and the fact that so many organizations already adopted it into their production environments. When I started drafting my [GenerativeAI talk](#), I interacted with OpenTelemetry, APM and Elastic's Observability stack. Seeing how everything blended together was a great experience, so when the chance opened up to contribute to OpenTelemetry project, I jumped right into it.

[OTCA certification from Linux Foundation](#) has been released on the market just very recently. It is the first exam and certification covering the OpenTelemetry topic at its foundational level. I got the chance to contribute as a developer to OTCA exam, writing questions, double checking the documentation sources and putting myself in the shoes of a person who wants to get certified on OpenTelemetry. For me it has been the perfect opportunity to get closer to an open source project I admired from afar and interacting with the open source community.

Educate through written content

Document writing: a great way to contribute to Open Source projects



Diana Todea · 2 min read · Feb 20, 2025

This year I have started further contributing to Open Source projects, in my case this was [OpenTelemetry](#). OTel has a great need for documentation translation in various languages, you can check their announcement [here](#).

Since my favorite Open Source project needed help I decided to add my contribution helping translating documentation for the [Spanish website](#).

We are just getting started so the need for contributors is acute. It doesn't necessarily take to write code in order to contribute to an open source project, it can be a small step as helping translating a page.

All the steps to start contributing to site localization in different languages is explained in this [directory](#).

In this short article I wanted to share my experience so far as a first time contributor to OTel documentation. It was been an easy onboarding from the first moment and I received support from the maintainers and other contributors of this site via the Slack channel: [#otel-docs-localization](#). Each

Take aways

- Diversify your contributions
- Distinguish between code and no code contributions
- Even the tiniest contribution matters to the community
- Prioritize and nurture local communities and docs localization projects
- Crosspost your findings on different channels
- Pay it forward: keep the collaboration open between OSS projects

We need your contributions!



#otel-profiles

#otel-system-metrics

#otel-contributor-experience



Join CNCF Slack



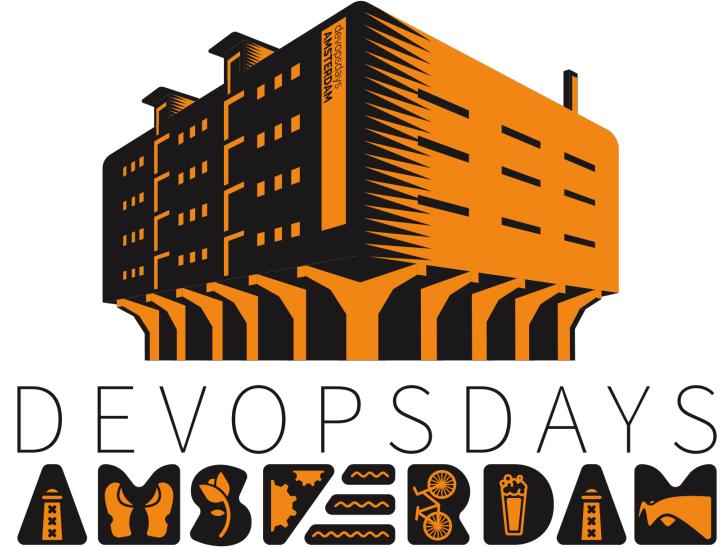
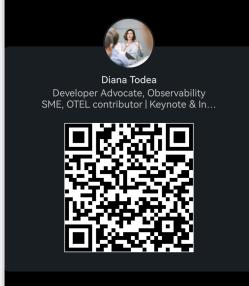
#otel channels

- I played with OTel and I liked it
- Why feedback matters for Open Source
- Contribute to OTel (OTel blog)

- <https://opentelemetry.io/blog/2025/devex-survey>
- <https://opentelemetry.io/blog/2025/contribex-survey-results/>
- <https://training.linuxfoundation.org/certification/opentelemetry-certified-associate-otca/>

THANKS Q&A

[github@didiviking/Conferences_Talks](https://github.com/didiviking/Conferences_Talks)



2025