

DORA Metrics in Practice: Make them Observable through your Platform!



Danielle Cook
CNCF Ambassador



Andreas Grabner
CNCF Ambassador and DevRel
Dynatrace



1: New to DORA

I am here to learn what it's all about.

2: Understand DORA

I am here to learn how to implement it.

3: Implemented DORA

I am here to learn what else is possible.

4: DORA is lie

I am here because I haven't seen it work in practice!

Why DORA Metrics?

...“aims to understand what **differentiates high-performing** DevOps teams from **low-performing ones**, and to provide insights and metrics that can **help organizations improve their software delivery and operations.**”



2024 DORA report



4 Key DORA Metrics



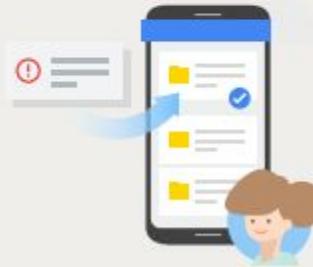
Change lead time:
the time it takes for a code commit or change to be successfully deployed to production.



Deployment frequency:
how often application changes are deployed to production.



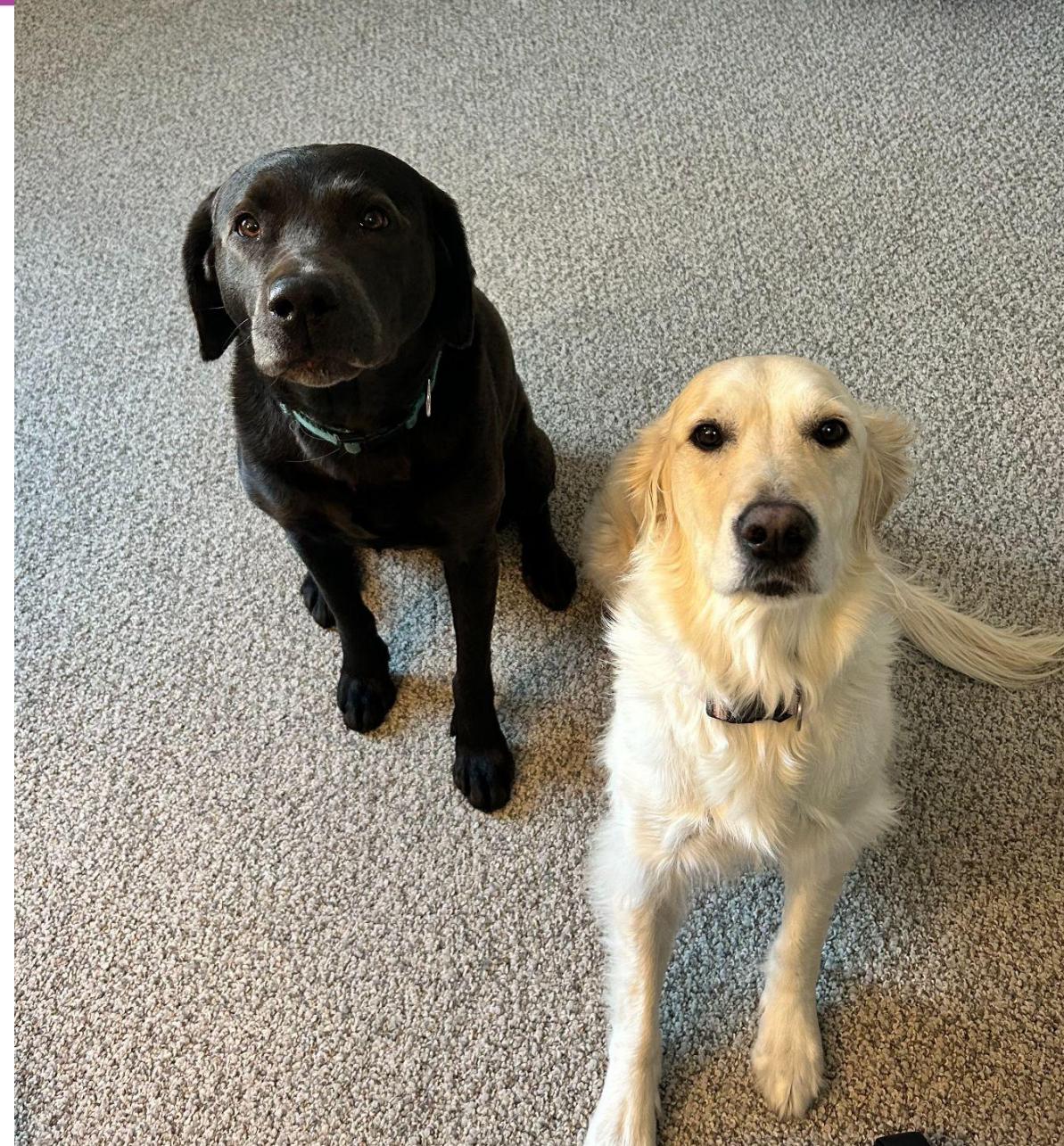
Change fail rate:
the percentage of deployments that cause failures in production,¹ requiring hotfixes or rollbacks.



Failed deployment recovery time:
the time it takes to recover from a failed deployment.

DORA Metrics Can Be Hard to Track

- Most teams **don't have a clear view** of their DORA metrics.
- Data is **scattered** across CI/CD, incident management, and monitoring tools.
- **Common gaps:** No real-time insights, manual reporting, limited visibility.
- The business **cares** (but might not know why).



CI/CD Observability enables DORA!



KubeCon

CloudNativeCon

Europe 2025



▽ ▽

How we streamlined our SDLC with Observability

from GitHub via Jenkins, Harbor, Argo to K8s and how the community can benefit from this!

PRESENTER
Michael Gläss
Chief Product Architect

PRESENTER
Andreas Grabner
CNCF Ambassador and DevRel



How we made OpenTelemetry Be Our Fitness Tracker for CI/CD Pipelines

Through the power of OpenTelemetry

PRESENTER
Nicolas Woerner
CLARIO.

PRESENTER
Andreas Grabner
CNCF Ambassador
dynatrace

BLOG / SIG POST

OpenTelemetry Is expanding into CI/CD observability

Posted on November 4, 2024 by Dotan Horovits + Adriel Perkins

CNCF projects highlighted in this post



SIG post by [Dotan Horovits](#) and [Adriel Perkins](#), Project Leads, SIG CI/CD Observability, OpenTelemetry

We've been talking about the need for a common "language" for reporting and observing CI/CD pipelines for years, and finally, we see the first "words" of this language entering the "dictionary" of observability – the OpenTelemetry open specification. With the recent release of OpenTelemetry's Semantic Conventions, v1.27.0, you can find [designated attributes for reporting CI/CD pipelines](#).



Who is building a platform for a development organization of the size

1: Smaller than 100

2: Between 101 and 1000

3: For more than 1000

Case Study: Dynatrace

2000

Developers

300

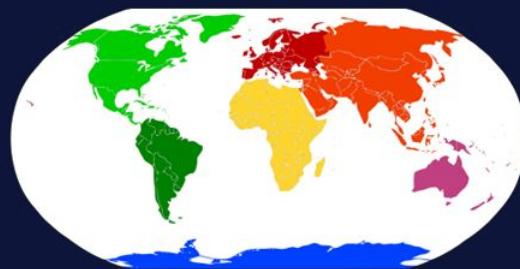
Builds / Day

250+

Services / Stage



Cloud Vendors



On every continent

Tech Stack: Cloud Native mixed with Managed Cloud Services

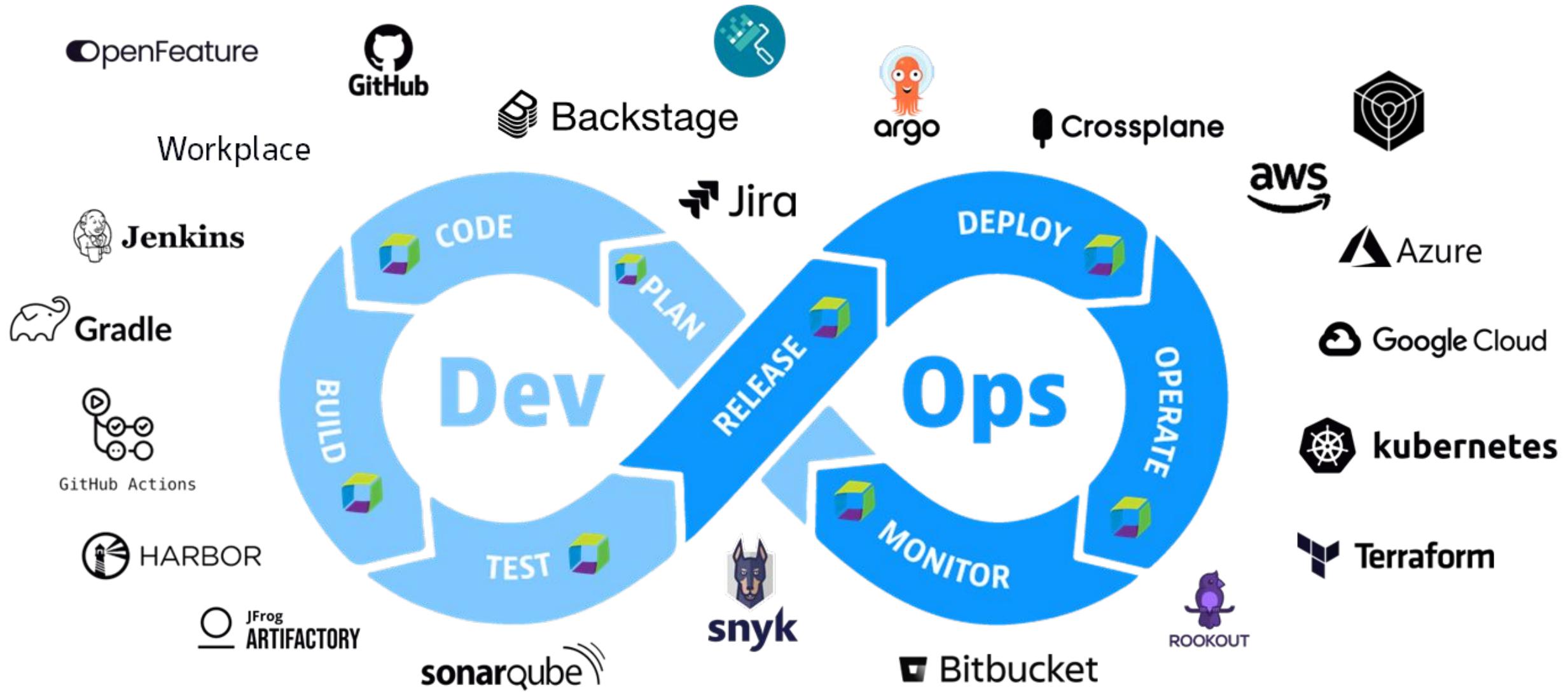
**7+2+2 dev,
3x3 hardening,
3 bugfix, X prod**

Stages

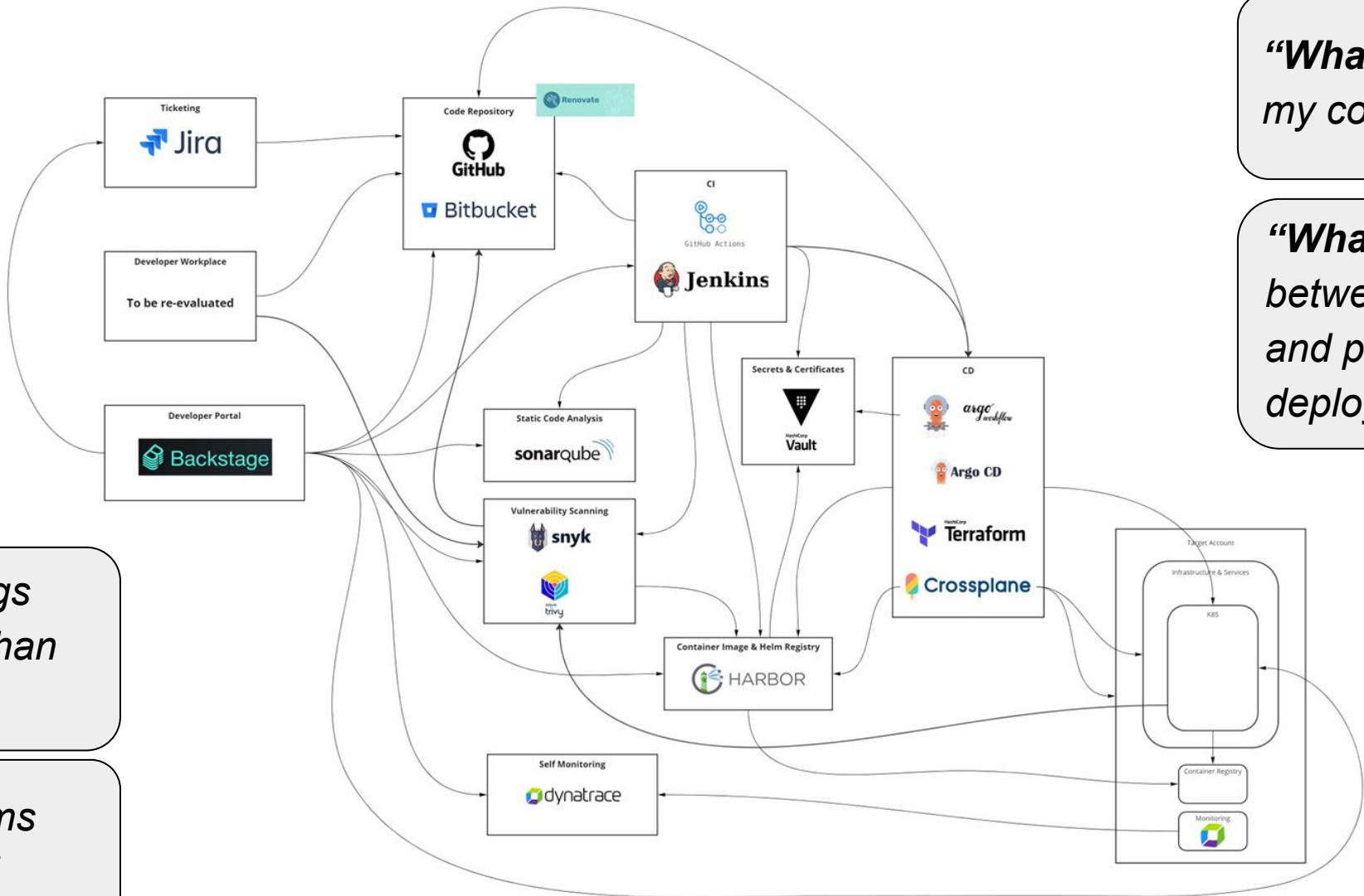
Every 2 weeks
Prod-Rollouts

SDLC (Software Delivery Lifecycle)

Lots of tools involved from git commit to prod

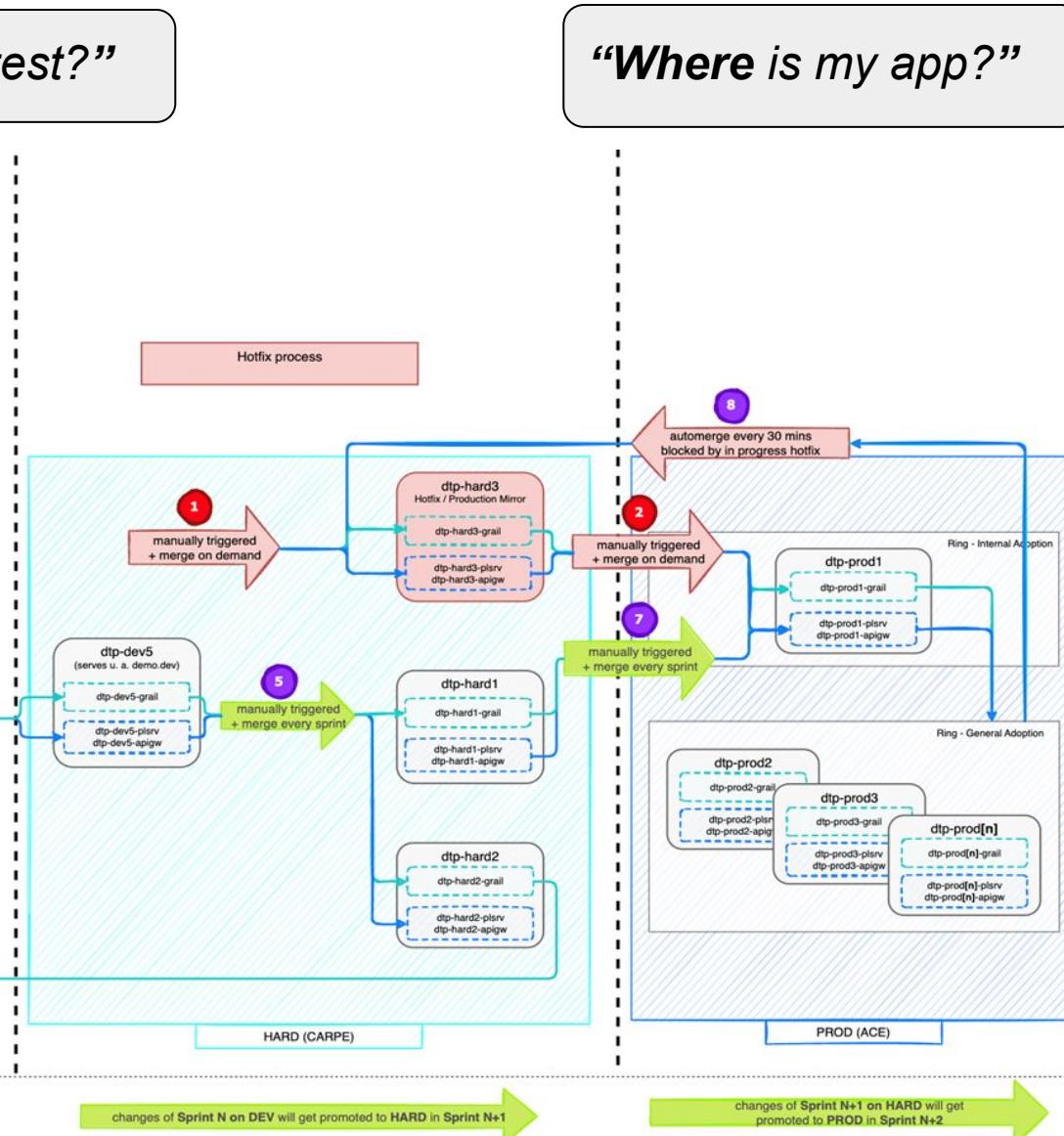
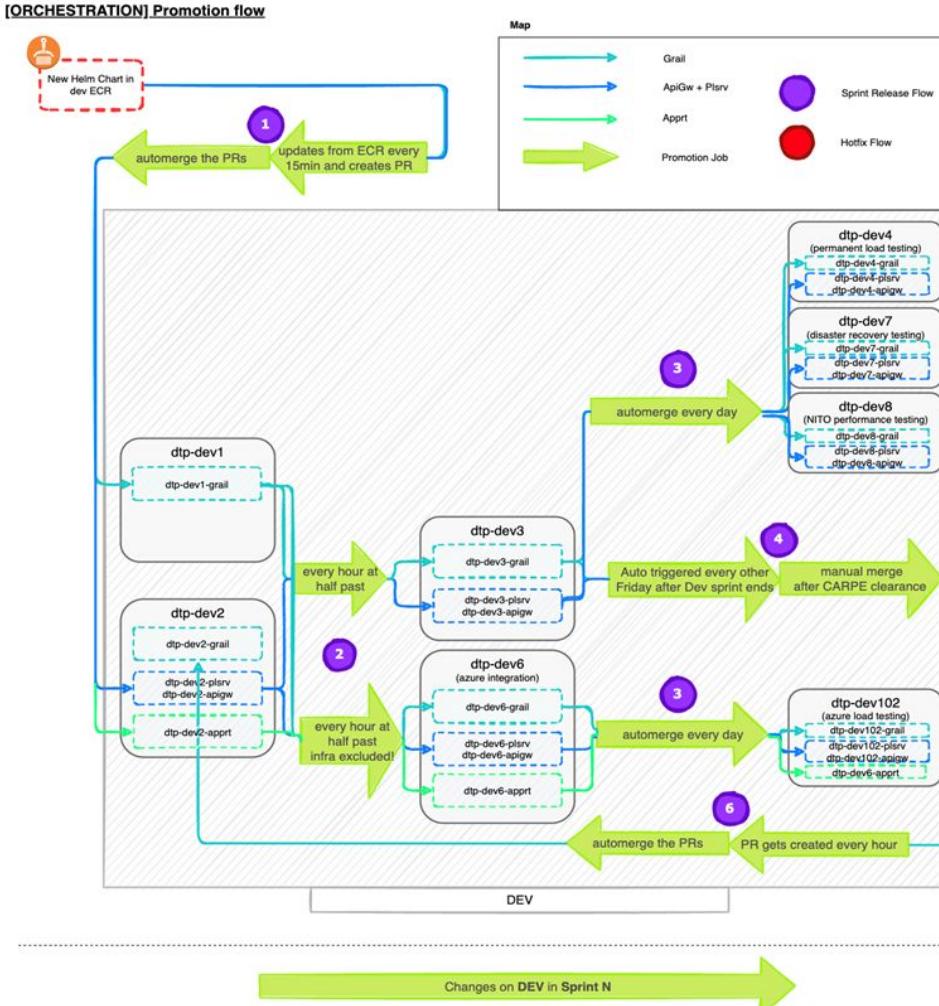


Natural Challenges based on our evolved complex “Tooing Landscape”



Natural Challenges In staging and orchestration

“Why is my service still in development and not in test?”



JUNO - Developer Platform @ Dynatrace

To streamline development of new Dynatrace Apps



This section provides a detailed look at the JUNO developer platform's integrated toolchain:

- Workflow Actions:** Shows a Jira card for a "Product Management / Workflow Action".
- Create a new component:** Shows the Dynatrace Backstage interface for creating software components.
- Source:** Shows the Dynatrace DevContainer interface for managing source code repositories (e.g., GitHub, Bitbucket).
- New:** Shows the Dynatrace DevContainer interface for creating new workspaces.
- Service Health Score Card:** Shows the Dynatrace Debugger interface displaying a service health score card and execution history.

The Dynatrace logo is centered at the bottom, with a dotted line extending upwards from its base. To the right of the logo, there is a row of logos for various cloud providers and services:

- AWS Lambda
- Azure
- Google Cloud

Tool & System Observability

Apply the same best practices as for any critical app

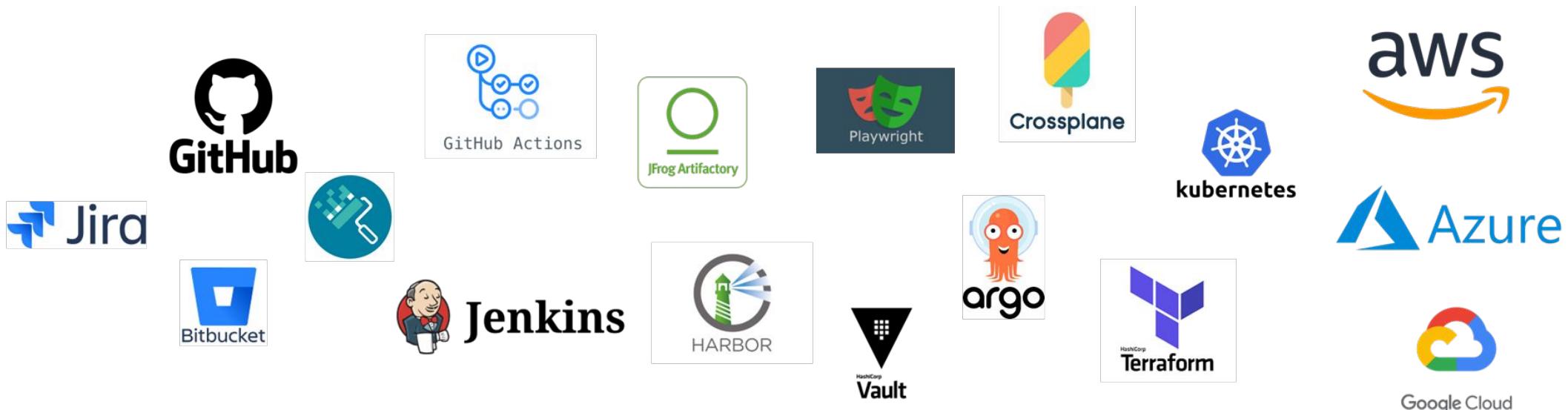


KubeCon
Europe 2025



CloudNativeCon
Europe 2025

"Every tool and system that is involved in the DevOps cycle needs to get observed and the data collected towards a single source of truth."



Evolution: From Tool to Pipeline Observability

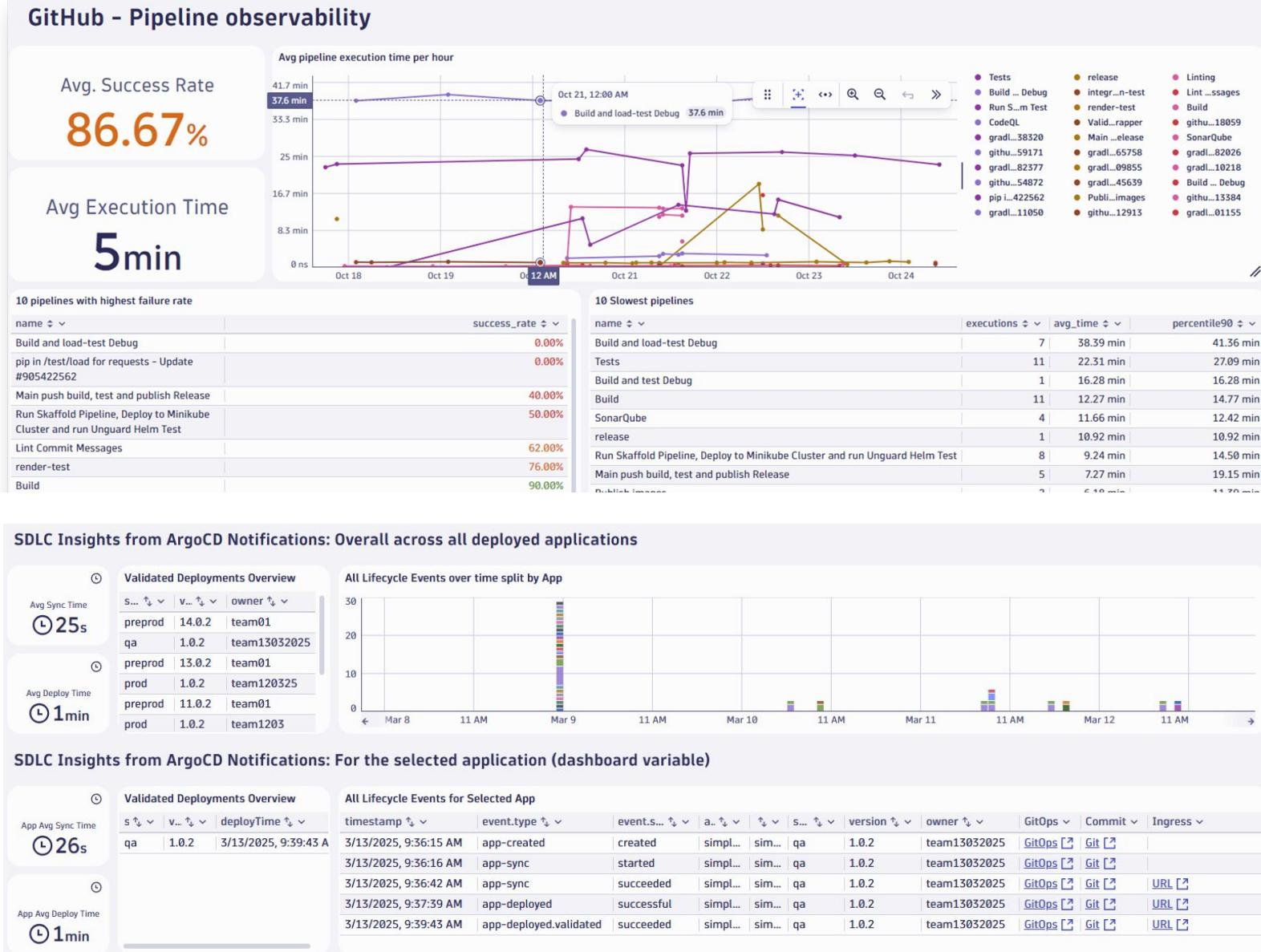
Extracting Metrics & Traces from Git* Webhooks



Webhooks



Notifications



Analyze Deployment Overview



KubeCon CloudNativeCon
Europe 2025

*As a next step we created a **comprehensive deployment overview** in order to see **what is running where**.*

How do you know what systems, processes and jobs are involved into your DevOps Process?

Feature Branch		Release Branch					
Local Development		Automated Software Build		Automated Container Build		Tests	
Code Change	Push Code	Code Build		Container Build		Integration Tests	Merge
		(Unit) Tests		Component Tests		Automated E2E Tests	Merge Validation
		Code Vulnerability checks		Container Vulnerability checks			
		Code Quality checks	OK/NOK	SBOM	OK/NOK	Deploy to Sandbox	Pull Request
		SBOM		Signing		API / Contract Test	OK/NOK Approval?
		Signing		Container upload		Load Testing	
		Library / SDK upload					

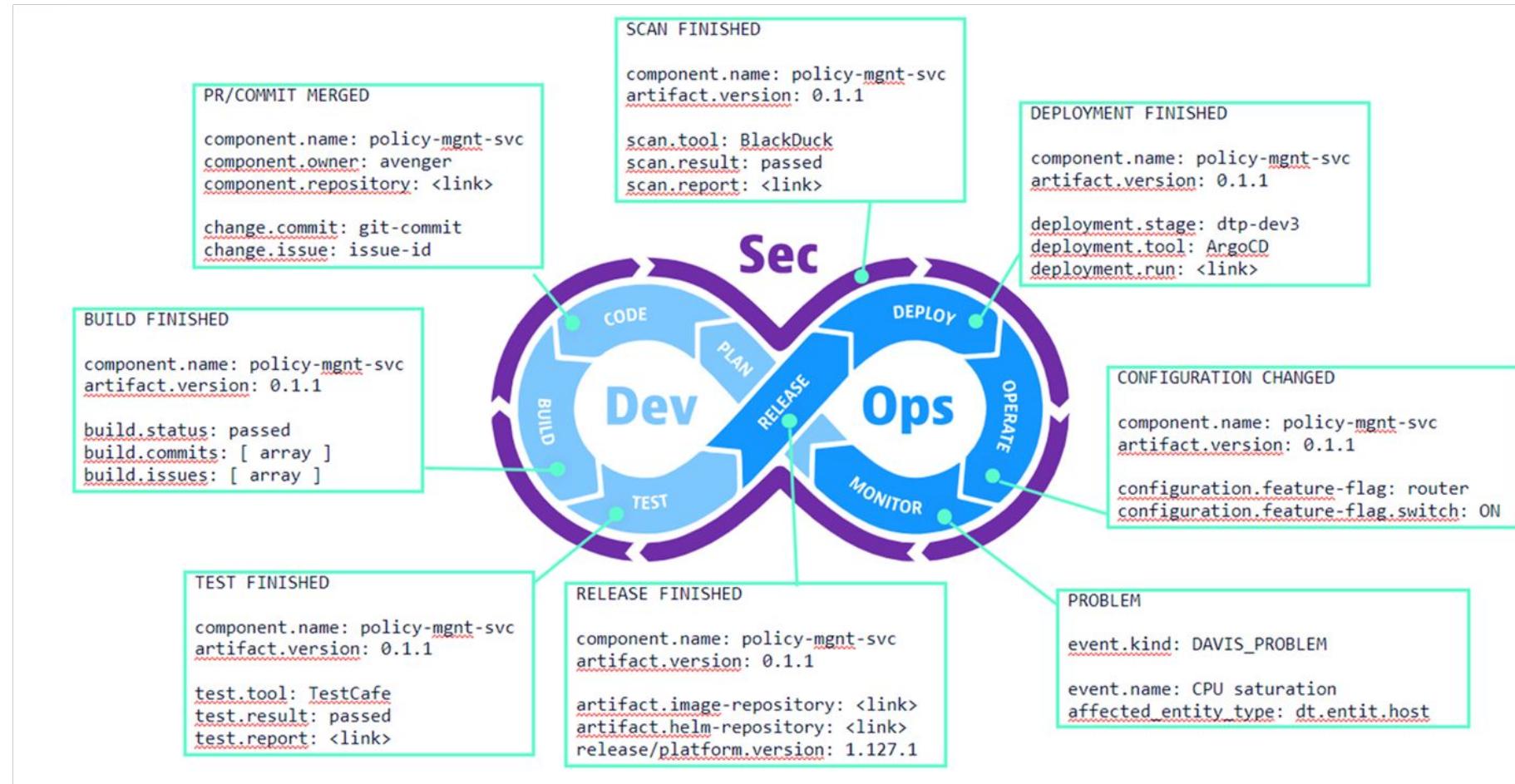
*Getting an **event** from every of these steps is a first collection helping to analyze the flow.*

As soon as you are jumping tool boundaries you need a key that individually connects caller with executer.

Leading to SDLC (Software Delivery Lifecycle) Events

From first git commit to deployment to operation

All jobs or actions need to trigger **SDLC Events with relevant information** to enrich the system observability



We embedded SDLC Events into our Sematic Dictionary to ensure consistent usage.

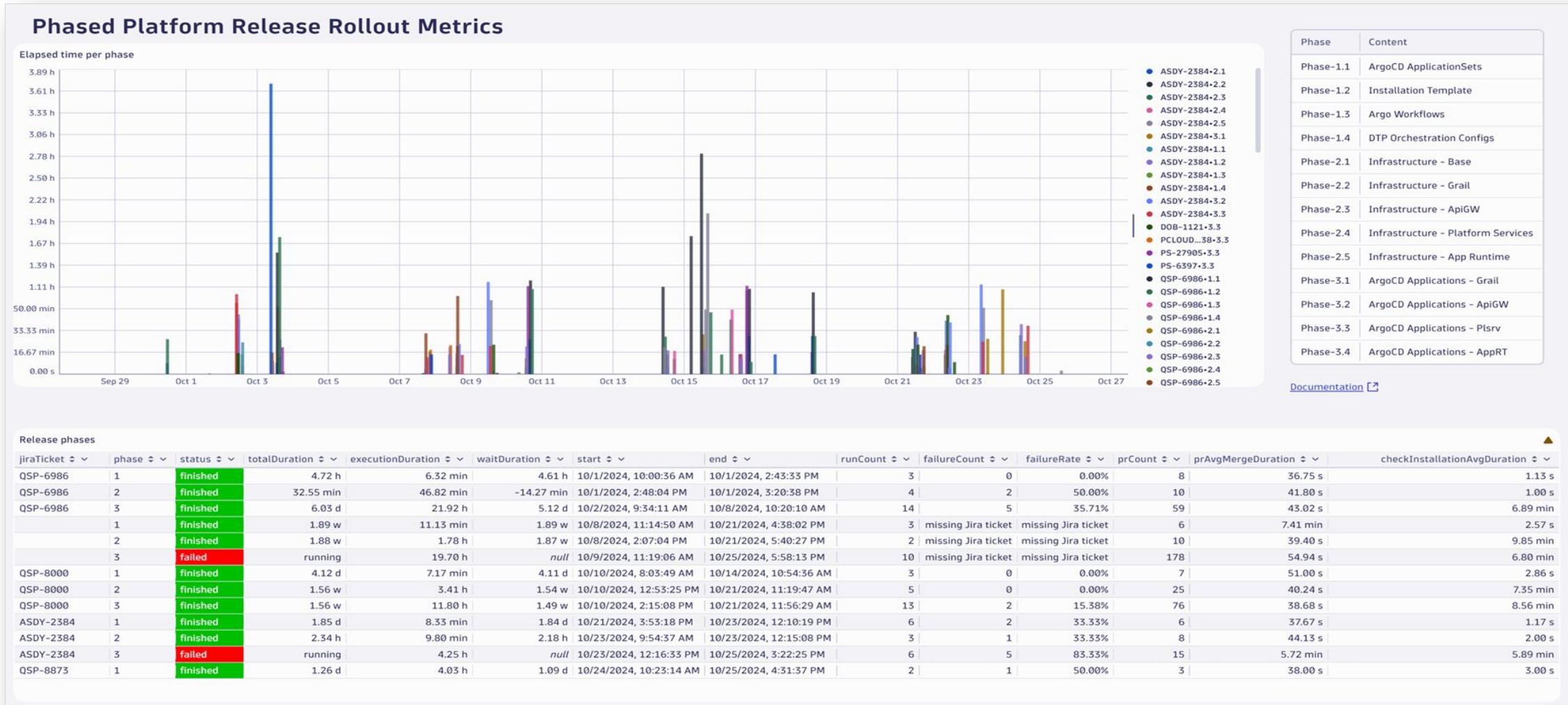
Our Delivery Quality Metrics

Insights into deployment phases!



KubeCon CloudNativeCon
Europe 2025

We built a **visualization** of our automated **rollout processes** with it's phases and **relevant measures**.



BLOG / SIG POST

OpenTelemetry Is expanding into CI/CD observability

Posted on November 4, 2024 by Dotan Horovits + Adriel Perkins

CNCF projects highlighted in this post



SIG post by [Dotan Horovits](#) and [Adriel Perkins](#), Project Leads, SIG CI/CD Observability, OpenTelemetry

We've been talking about the need for a common "language" for reporting and observing CI/CD pipelines for years, and finally, we see the first "words" of this language entering the "dictionary" of observability – the OpenTelemetry open specification. With the recent release of OpenTelemetry's Semantic Conventions, v1.27.0, you can find [designated attributes for reporting CI/CD pipelines](#).

OpenTelemetry: Semantic Conventions for CI/CD



KubeCon
Europe 2025



CloudNativeCon
Europe 2025

- **Standardized dimensions for metadata**
- Ensure **consistency** in how data is interpreted across **different systems**

Attribute	Type	Description	Examples	Stability
cicd.pipeline.name	string	The human readable name of the pipeline within a CI/CD system.	Build and Test ; Lint ; Deploy Go Project ; deploy_to_environment	experimental
cicd.pipeline.result	string	The result of a pipeline run.	success ; failure ; timeout ; skipped	experimental
cicd.pipeline.run.id	string	The unique identifier of a pipeline run within a CI/CD system.	120912	experimental
cicd.pipeline.run.state	string	The pipeline run goes through these states during its lifecycle.	pending ; executing ; finalizing	experimental
cicd.pipeline.task.name	string	The human readable name of a task within a pipeline. Task here most closely aligns with a computing_process in a pipeline. Other terms for tasks include commands, steps, and procedures.	Run GoLang Linter ; Go Build ; go-test ; deploy_binary	experimental
cicd.pipeline.task.run.id	string	The unique identifier of a task run within a pipeline.	12097	experimental
cicd.pipeline.task.run.url.full	string	The URL of the pipeline run providing the complete address in order to locate and identify the pipeline run.	https://github.com/open-telemetry/semantic-conventions/actions/runs/9753949763/job/26920038674?pr=1075	experimental
cicd.pipeline.task.type	string	The type of the task within a pipeline.	build ; test ; deploy	experimental
cicd.system.component	string	The name of a component of the CICD system.	controller ; scheduler ; agent	experimental
cicd.worker.state	string	The state of a CICD worker / agent.	idle ; busy ; down	experimental

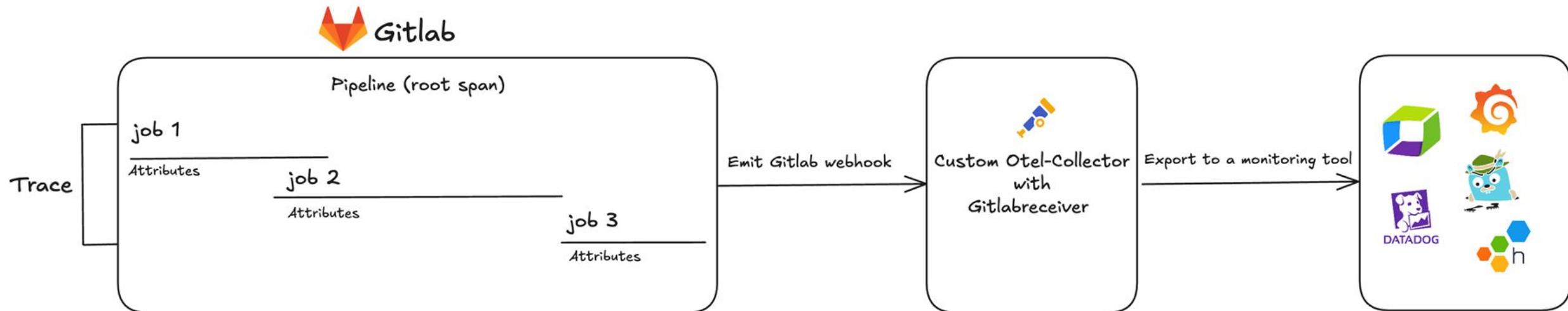
Tracing GitLab Pipelines



KubeCon
Europe 2025



CloudNativeCon
Europe 2025



How we made OpenTelemetry
Be Our Fitness Tracker for
CI/CD Pipelines

Through the power of OpenTelemetry

PRESENTER
Nicolas Woerner
CLARIO.

PRESENTER
Andreas Grabner
CNCF Ambassador
dynatrace

Tracing Every Step of your Pipelines with OTEL

Spotting where things slow down or fail!



Trace: Gitlab Pipeline: 1525031164 - <https://gitlab.com/clario-clinical/shared/personal/niwoerner/pipeline-observeability/salt-lake-city-service/-/pipelines/1525031164>

Close details

⌚ Duration: 16,23 min ⚖ Response time: 16,23 min | 📅 3. November 2024 um 16:09:15 | Service: [clario-clinical/shared/personal/niwoerner/pipeline-observeability/salt-lake-city-service](https://gitlab.com/clario-clinical/shared/personal/niwoerner/pipeline-observeability/salt-lake-city-service)

Trace ID: b94fc9227bff576314d2b6a3fa9b8b9e

12 spans Search name, endpoint, service, or attributes



Putting your DORA metrics on a dashboard

Deployment Frequency, Lead Time, Failures



Performance level	Change lead time	Deployment frequency
Elite	Less than one day	On demand (multiple deploys per day)
High	Between one day and one week	Between once per day and once per week
Medium	Between one week and one month	Between once per week and once per month
Low	Between one month and six months	Between once per month and once every six months

All links in a spot



▽ ▽

How we streamlined our SDLC with Observability

from GitHub via Jenkins, Harbor, Argo to K8s and how the community can benefit from this!



PRESENTER
Michael Gläss
Chief Product Architect

PRESENTER
Andreas Grabner
CNCF-Ambassador and DevRel



How we made OpenTelemetry Be Our Fitness Tracker for CI/CD Pipelines

Through the power of OpenTelemetry



PRESENTER
Nicolas Woerner
CLARIO.

PRESENTER
Andreas Grabner
CNCF Ambassador
dynatrace

BLOG / SIG POST

OpenTelemetry Is expanding into CI/CD observability

Posted on November 4, 2024 by Dotan Horovits + Adriel Perkins

CNCF projects highlighted in this post



SIG post by [Dotan Horovits](#) and [Adriel Perkins](#), Project Leads, SIG CI/CD Observability, OpenTelemetry

We've been talking about the need for a common "language" for reporting and observing CI/CD pipelines for years, and finally, we see the first "words" of this language entering the "dictionary" of observability – the OpenTelemetry open specification. With the recent release of OpenTelemetry's Semantic Conventions, v1.27.0, you can find [designated attributes for reporting CI/CD pipelines](#).



Aligning Platform Value to Business Value

DORA Metric	Business Metric	Business Impact
Change Lead Time (LT)	Time to Value, Time to Market, Customer Satisfaction	Faster feature delivery leads to better customer experience.
Deployment Frequency (DF)	Marketing & Sales Pipeline, Customer Retention	Enables continuous innovation, supporting go-to-market teams, retention rates.
Change Failure Rate (CFR)	Operational Efficiency, Tooling Costs, Team Growth	Lower failure rates reduce rework, saving costs and time.
Failed Deployment Recovery Time (MTTR)	Revenue Protection, Brand Trust, Customer Churn	Faster recovery minimizes downtime impact on customers.



DORA Metrics in Practice: Make them Observable through your Platform!



Danielle Cook
CNCF Ambassador



Andreas Grabner
CNCF Ambassador and DevRel
Dynatrace

