What is CI/CD observability and how to bring it to CI/CD pipelines

A reliable CI/CD pipeline is the backbone of every modern project, yet there's limited visibility into its processes, often requiring manual review and analysis of build outputs. By leveraging OpenTelemetry standards we want to bring observability to our pipelines, making the software delivery process fully observable. Join our journey redefining CI/CD observability, learn how you can start implementing the techniques we are using at Grafana Labs to ensure your pipelines are more reliable and stay performant over time, how to identify flakiness, bottlenecks, and how we envision a future where - no matter your system or your observability solution - we can effortlessly have full visibility over our software delivery process. You will learn what it takes to make CI/CD fully observable, how we think OpenTelemetry is going to play a major role in this, what obstacles we encountered, what are the challenges ahead and how anyone can help shape the future of CI/CD observability.
What is CI?
- Current issues with CI/CD systems
- Intro to OpenTelemetry
- Semantic Conventions
- Own your data
- Practical use cases for CI/CD observability
- What’s next?
What is CI?
Continuous Integration: The continuous assembling and testing of complex and rapidly evolving systems.

Continuous Integration: The process of combining code changes frequently, with each change verified on check-in.
What is CI?
But like - for real
Definition of Continuous Integration but for real

A mechanism to:

- Reduce risks
- Reduce repetitive manual processes
- Generate deployable software at any time and at any place
- Enable better project visibility
- Establish greeted confidence in the software product
- Find and resolve flaky tests/builds
- ...

...and of course (try to) prevent paging people at 3AM :-(
CI/CD PIPELINE

TEST
WAIT FOR CHANGE APPROVAL
MORE TWEAKS
HEAT DEATH OF UNIVERSE
EVERYTHING WORKS!
JUST KIDDING: WHY WOULD ANY OF THIS WORK
ERRORS
DOWNTIME
PANIC
ADD LINKEDIN SKILLS: DEVOPS, AUTOMATION, TROUBLESHOOTING

BUILD
MORE TWEAKS
HEAT DEATH OF UNIVERSE
DEPLOY TO PROD
50-PERSON CONFERENCE CALL
MANUAL TWEAKS
DEPLOY STAGING
CHANGE APPROVED SIGHT-UNSEEN
PROD RETURNS TO AN ACCEPTABLY BROKEN STATE

DEPLOY DEV
NOTHING WORKS
DEPLOY TO PROD
WAIT FOR 3AM MAINTENANCE WINDOW

MANUAL TWEAKS
DEPLOY STAGING
FORGET ABOUT THE TWEAKS

What is CI?
But like – for **real**
  real
  real
Definition of Continuous Integration for real real

CI and Alerting: A Common Purpose

- Proactive issue identification
- Continuous system monitoring
- Rapid response to anomalies/outliers

CI as the "Left Shift" of Alerting

- Early detection in development
- Preemptive issue resolution
- Shifting focus to proactive monitoring
Continuous Integration (CI)

- Early Detection and Issue Catching in Development
- Build Health and System Maintenance by utilizing signals
- Continuous Monitoring for System Health
- Implementing Actionable Alerts and Tests
- Viewing CI and Alerting as Complementary Components
Continuous Integration (CI)

- Early Detection and Issue Catching in Development
- Build Health and System Maintenance by utilizing signals
- Continuous Monitoring for System Health
- Implementing Actionable Alerts and Tests
- Viewing CI and Alerting as Complementary Components

Alerting

- Rapid Problem and Timely Issue Identification
- Service-Level Objective (SLO) Compliance using signal monitoring
- Well-Managed Alerting for System Health
- Dealing with Brittle Alerts and Tests
- Overcoming Siloed Perspectives
Current issues with CI/CD systems
Observability so far

- Local runs - fmt.Println("here") (choose your fav language)
- “Let’s page the Platform team, they should know!”
- Create alerts after the fact
- Github → CI check webhook → Grafana *

- Limited visibility during earlier stages
- Difficulty in root cause analysis
- Increased mean time to recovery (MTTR)
- Missed optimization opportunities
Observability so far
Observability so far

Source: https://www.pinterest.com/pin/this-is-not-fine--37295503147896436/
Getting data out of CI/CD

FOCUS SHIFT

- Proactive Issue Resolution
- Streamlined Development Workflow
- Enhanced System Reliability
- Cost Reduction
Getting data out of CI/CD

FOCUS SHIFT

Build
Test
Deploy

Source: https://imgur.com/diZkDjj
OpenTelemetry 101 - What is OTel?

An **observability** framework designed to create and manage telemetry data such as **traces**, **metrics**, and **logs**.

[https://opentelemetry.io/docs/what-is-opentelemetry/](https://opentelemetry.io/docs/what-is-opentelemetry/)
OTel & CI/CD - The interesting bits

Semantic conventions

Own the data that you generate

Standard naming scheme for common telemetry data types

https://opentelemetry.io/docs/what-is-opentelemetry/
Semantic Conventions

By Signal Type
- Events
- Logs
- Metrics
- Resource
- Trace

By Area
- General
- Cloud Providers
- CloudEvents
- Database
- Exceptions
- ...
- Continuous Integration & Continuous Delivery

https://opentelemetry.io/docs/specs/semconv/
Gio’s totally reliable Go code
Yo! 😊 I wrote some "Totally reliable Go code"™ to get the data we needed out of Drone into Loki, Tempo and Mimir, can you check it out?

So cool! This is the most reliable code I’ve ever seen! 😎 Can you make it so it also sends Logs to **ElasticSearch** plz?

That’s great! Can we get tracing data out of GitHub Actions?

.kill me (666 kB)
Own your data
grafana-ci-otel-collector

OTE Collector

Drone Receiver → Processors

Loki Exporter

OTLP Exporter

Prometheus Remote Write Exporter
Practical examples

- [https://plugins.jenkins.io/opentelemetry/](https://plugins.jenkins.io/opentelemetry/)
  - Jenkins plugins that exports data via OTLP protocol
- [https://github.com/krzko/run-with-telemetry](https://github.com/krzko/run-with-telemetry)
  - Run commands and export results as trace data
- [https://github.com/grafana/grafana-ci-otel-collector](https://github.com/grafana/grafana-ci-otel-collector)
  - Generate Logs, Metrics, and traces from Drone Pipelines
CI/CD Observability in practice
CI/CD Observability in practice
What it unlocks

RED-like metrics (Rate, Errors, Duration)

DORA metrics

Code coverage stats over time

Flakiness detection

CI performance regression

Security

...A lot more
What’s next

- An OpenTelemetry working group focusing on defining CI/CD semantic conventions
- More receivers / components or tools to get data out of CI/CD systems

Join the discussion

  - CI/CD Observability channel on CNCF Slack
- [https://github.com/open-telemetry/oteps/pull/223](https://github.com/open-telemetry/oteps/pull/223)
  - OpenTelemetry enhancement proposal
Thank you

Want to chat?
We’ll be at the Grafana booth
k8 - level 1 ~15:40