



**DEVOPS
WORLD**
by CloudBees

COMMUNITY

Embracing observability in Jenkins with OpenTelemetry

Cyrille Le Clerc

Director of Product Management, Observability @ Elastic

Speaker

- Cyrille Le Clerc
- Product manager on Observability @ Elastic, including lead for OpenTelemetry
- Was product manager on Jenkins @ CloudBees, including creator & maintainer of the Jenkins Pipeline Maven Plugin

Agenda

- Yet Another Jenkins Monitoring Plugin?
 - CI/CD outages are hard to solve
 - Status of running Jenkins in production
 - Using observability for CI/CD
- Instrumenting Jenkins with OpenTelemetry
- Getting started with the Jenkins and OpenTelemetry
- What's next
 - Better Jenkins observability for admins and for developers
 - Observability for CI/CD, beyond Jenkins

COMMUNITY

Yet Another Jenkins Monitoring Plugin?

CI/CD outages are hard to solve

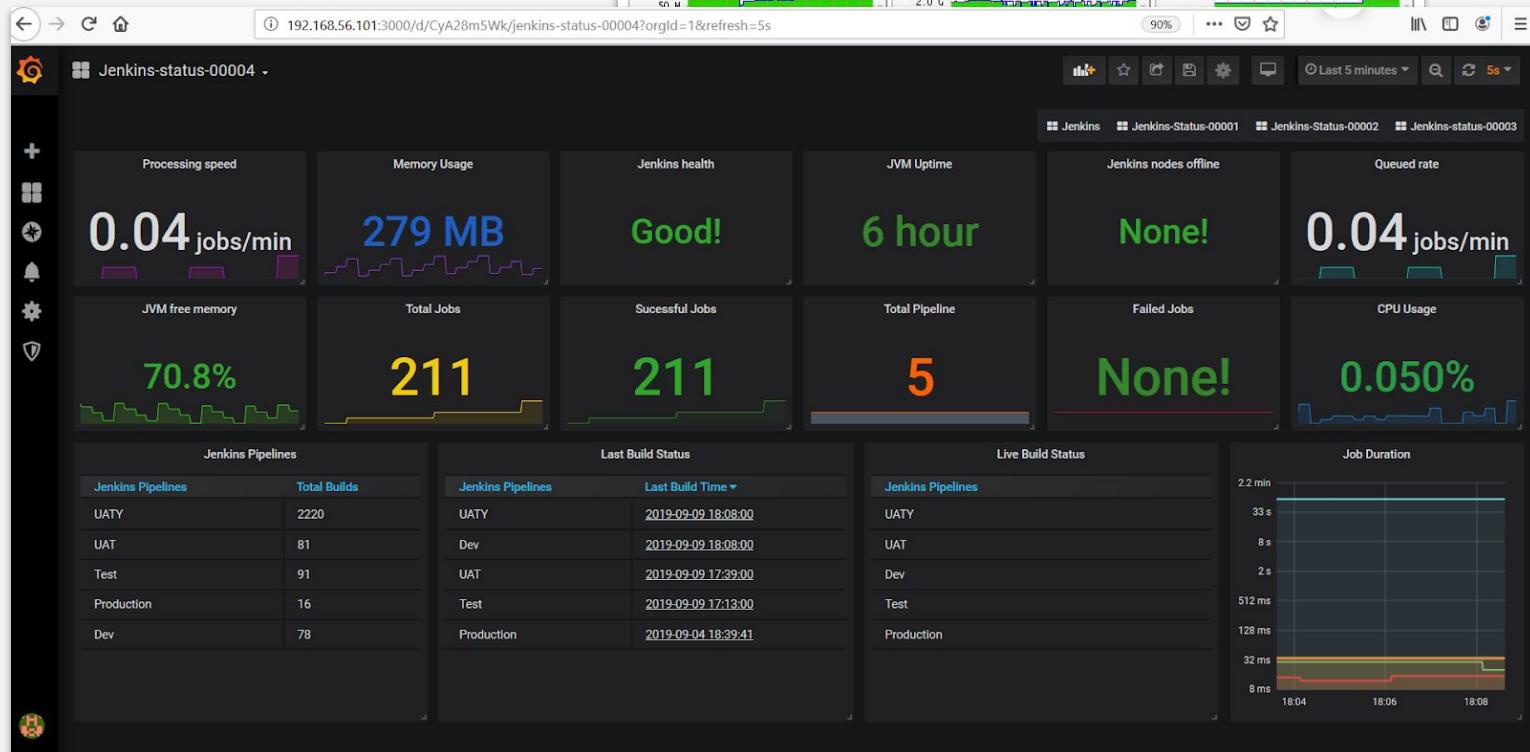
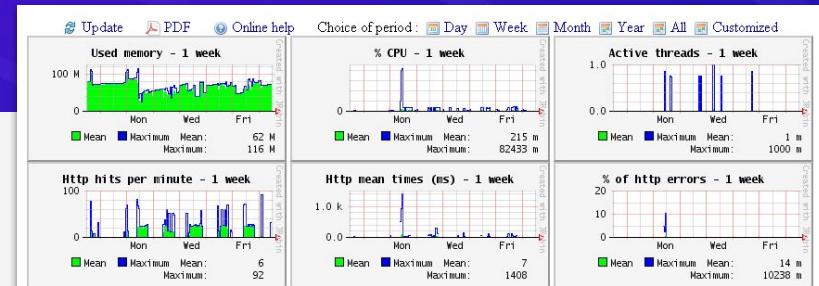
Status of running Jenkins in production

Using observability for CI/CD

Yet Another Jenkins Monitoring Plugin?

Jenkins Monitoring is a solved problem

- Jenkins Monitoring plugin (JavaMelody)
 - 20k installs
- Jenkins Metrics plugin
 - 35k installs, >75 metrics
- Jenkins Prometheus plugin
 - 8k installs
- Jenkins Datadog plugin
 - 2k installs

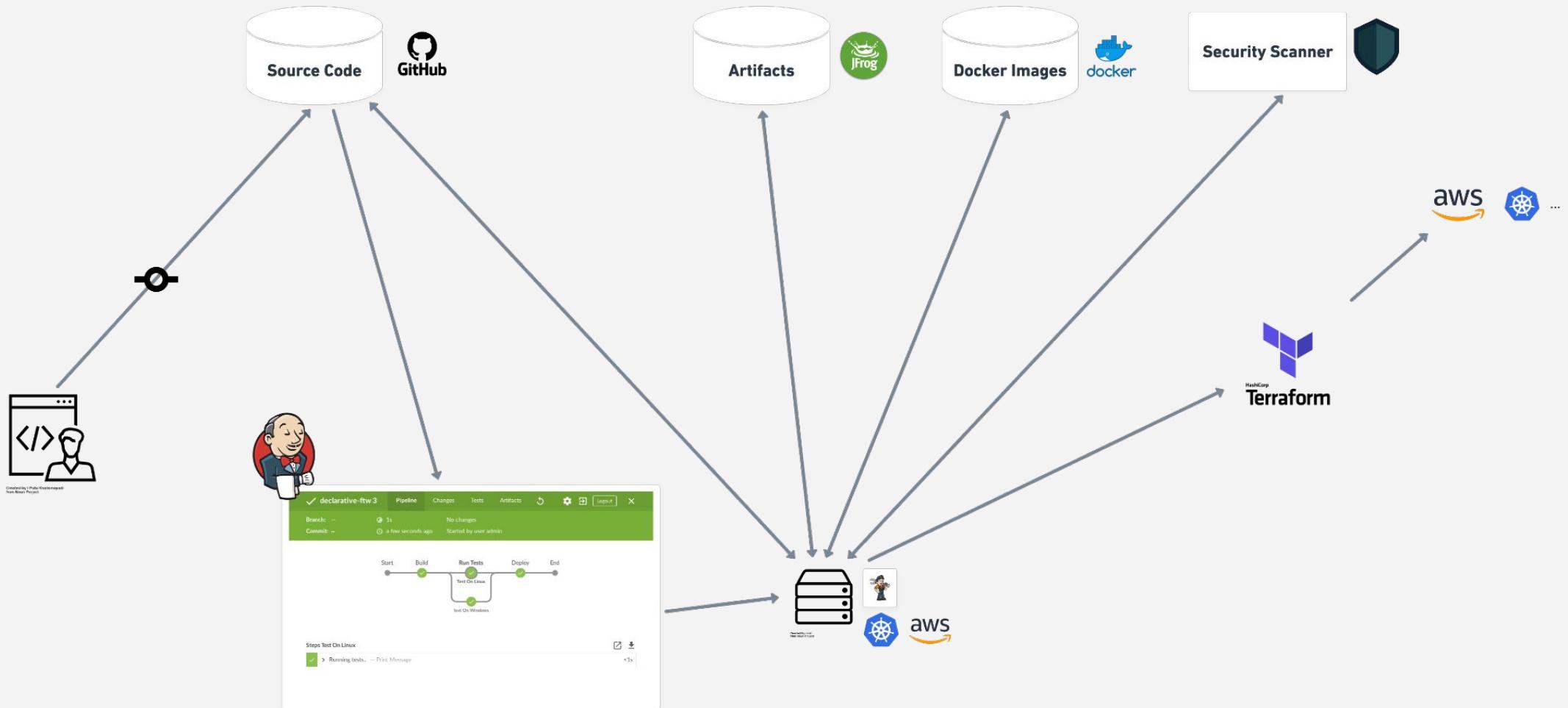


CI/CD outages are hard to solve

- Swiss Army knife connecting to everything
- Orchestrate things that fail like infrastructure and network communications
- Have high scalability problems with dynamic infrastructure
- Pipelines change all the time
- Are modified by many different personas with different levels of skills and knowledge on CI/CD platforms

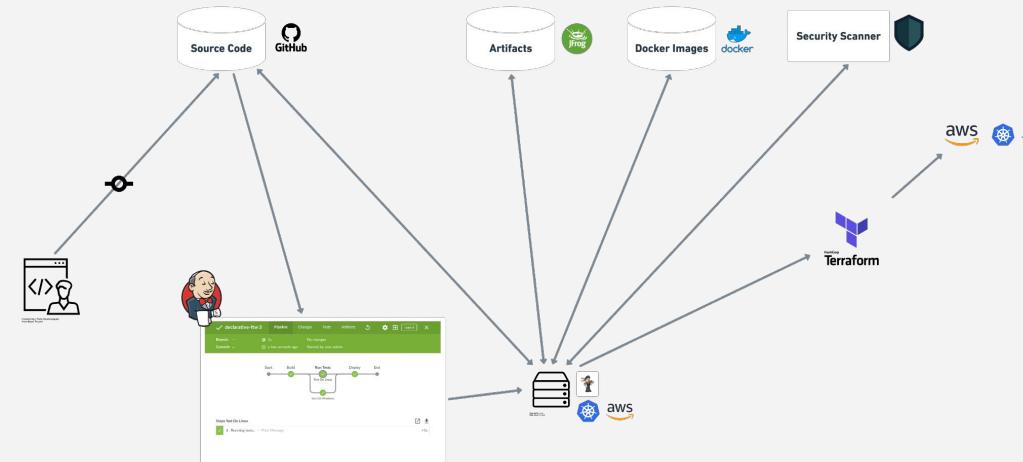


CI/CD outages are hard to solve



CI/CD outages are hard to solve

- Every integration will break one day
- Issue for a specific pipeline or for everybody
- Jenkins admins are often the last persons aware of problems



Status of running Jenkins in production

- Maintaining Jenkins up and running is a hard problem, both detection and root cause analysis are hard
- CI/CD is more critical than ever
- Jenkins admins are often the last persons aware of problems



Status of running Jenkins in production

- CI pipelines are by nature distributed architecture problems
- Lot of similarities with microservices and cloud native architectures who solved the problem

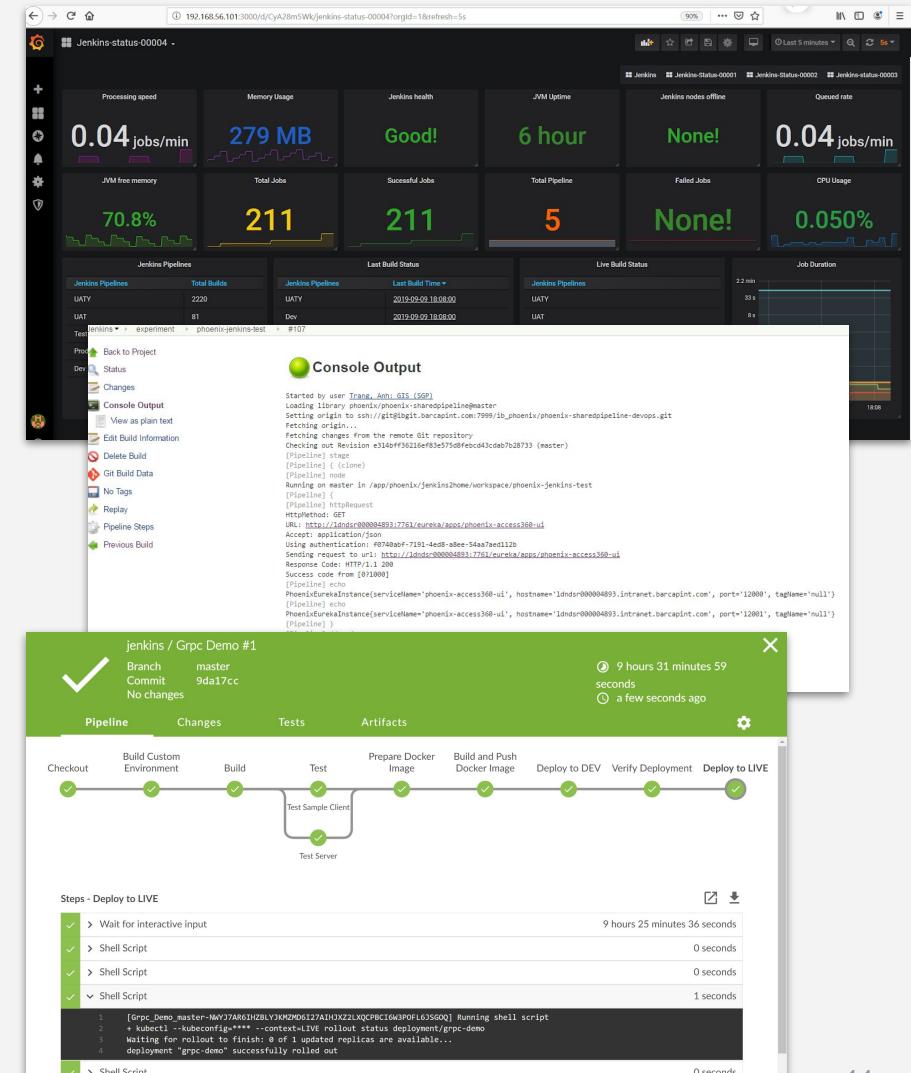


Credit https://commons.wikimedia.org/wiki/File:Rainbow_after_storm.jpg

Let's adopt in CI/CD the observability best practices and tools!

What is observability, monitoring vs observability

- Too many definitions of Observability
- Observability doesn't replace monitoring
- Observability is the capability to
 - Deep dive in problems
 - Not requiring to redeploy systems to add more troubleshooting info and details
 - Slicing and dicing in any dimension



Using observability for CI/CD

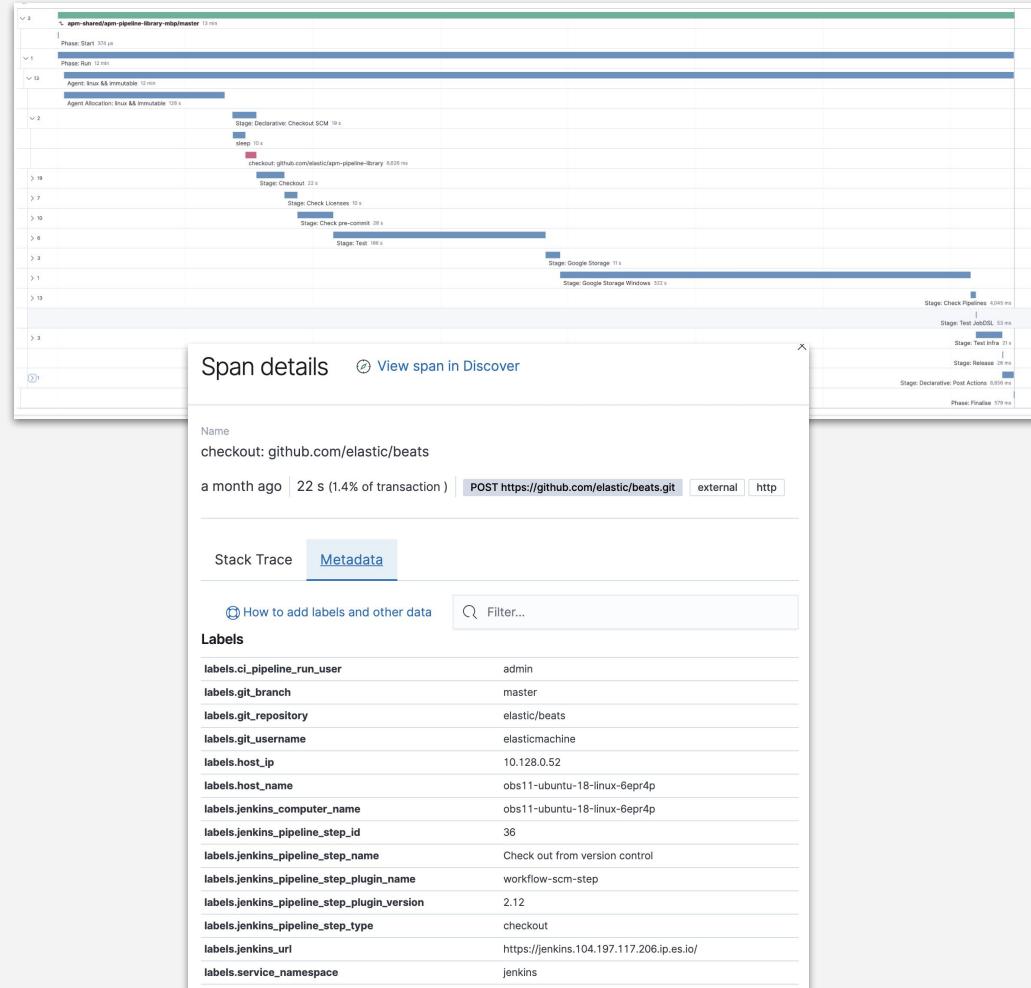
The image displays three screenshots illustrating the use of observability for CI/CD:

- Jenkins Pipeline Details:** Shows a Jenkins pipeline named "jenkins / Grpc Demo #1" with a green status bar indicating success. It lists steps: Checkout, Build Custom Environment, Build, Test, Prepare Docker Image, Build and Push Docker Image, Deploy to DEV, Verify Deployment, and Deploy to LIVE. The "Test" step has a sub-step "Test Sample Client" which is currently running. A "Console Output" window shows the logs for this step, including shell commands and their execution times.
- Span Details View:** A screenshot from a distributed tracing tool (like Jaeger) showing the execution of a Jenkins pipeline. The timeline shows various stages and their durations: Start (374 μs), Run (12 min), Agent allocation (12 min), Declarative: Checkout SCM (19 s), sleep (10 s), checkout: github.com/elastic/apm-pipeline-library (8,826 ms), Stage: Checkout (22 s), Stage: Check Licenses (10 s), Stage: Check pre-commit (28 s), Stage: Test (166 s), Stage: Google Storage (11 s), Stage: Google Storage Windows (322 s), Stage: Check Pipelines (4,045 ms), Stage: Test JobDSL (53 ms), Stage: Test Infra (21 s), Stage: Release (26 ms), Stage: Declarative: Post Action (8,804 ms), and Finalise (579 ms). A specific span for "checkout: github.com/elastic/beats" is highlighted, showing its duration of 22 seconds and its context as a POST request to https://github.com/elastic/beats.git.
- Jenkins Status Dashboard:** A screenshot of a Jenkins status dashboard showing various metrics. Key values include Processing speed (0.04 jobs/min), Memory Usage (70.8%), and JVM Uptime. It also shows Jenkins health, nodes offline, and queued rate. A "Console Output" window is also present at the bottom of the dashboard.

Using observability for CI/CD

Leverage capabilities of Observability tools

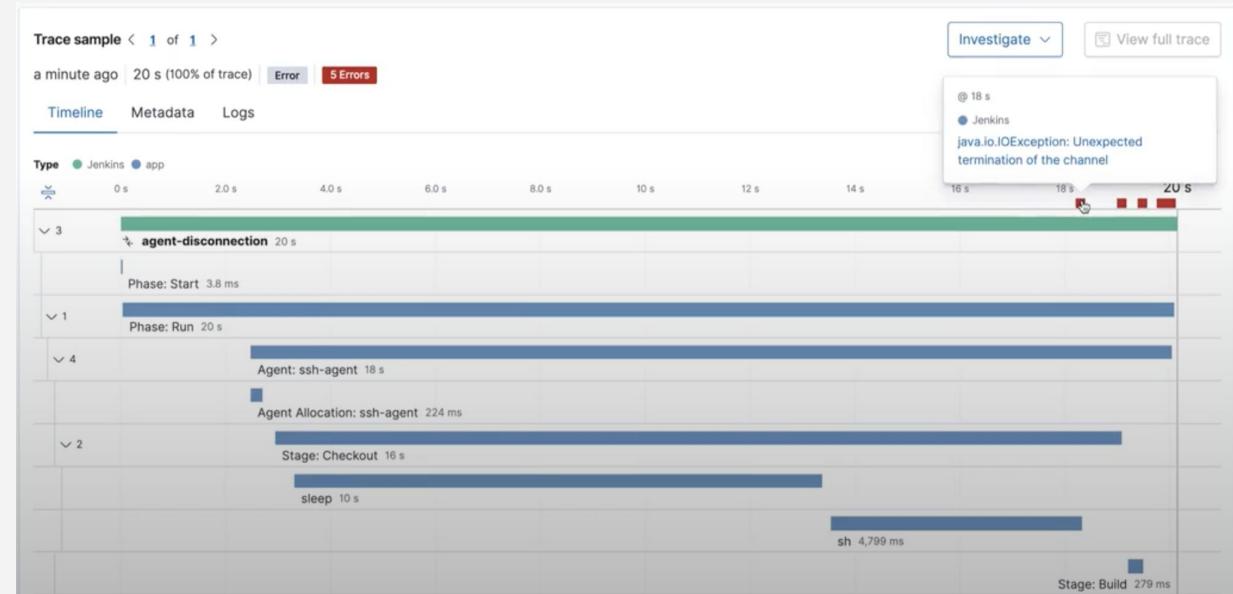
- Look at individual builds, at jobs, or globally
- Slice and dice in any dimension
- Use automated anomaly detection
- Use alerting



Using observability for CI/CD

Troubleshooting build agent issues

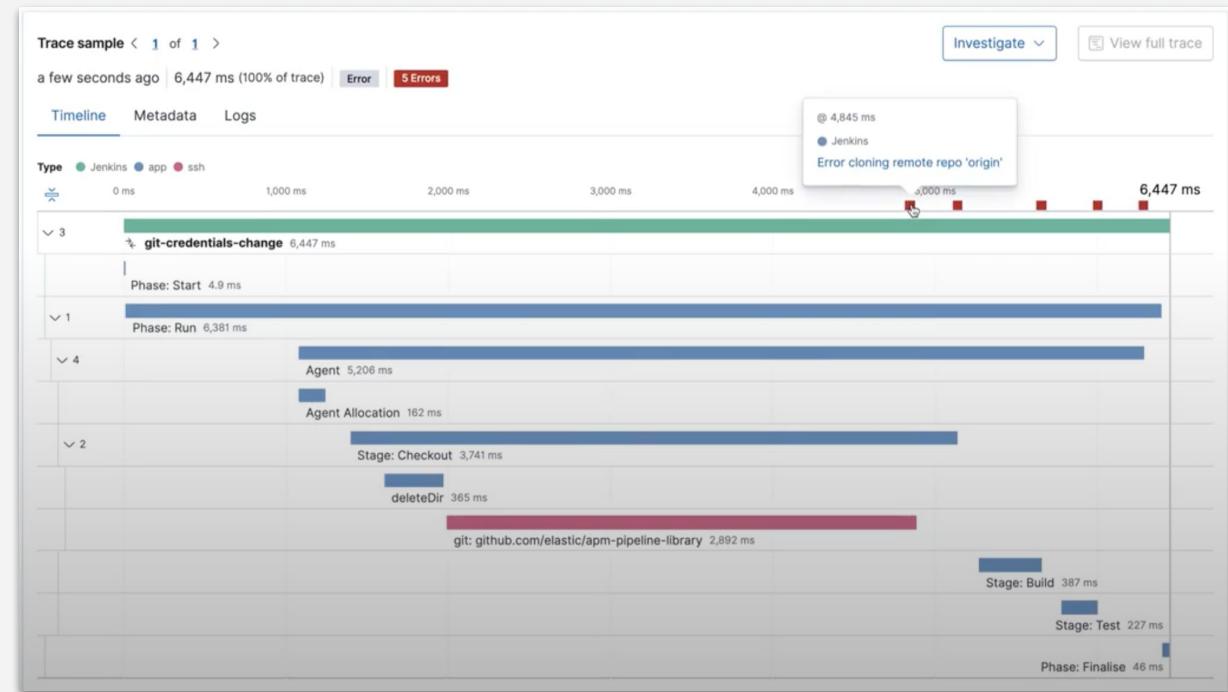
- E.g. network, dynamic provisioning...
- Impact only one pipeline, or everybody?



Using observability for CI/CD

Troubleshooting Git issues

- E.g. git server outage, credentials rotation, pipeline specific issue
- Impact only one pipeline, or everybody?

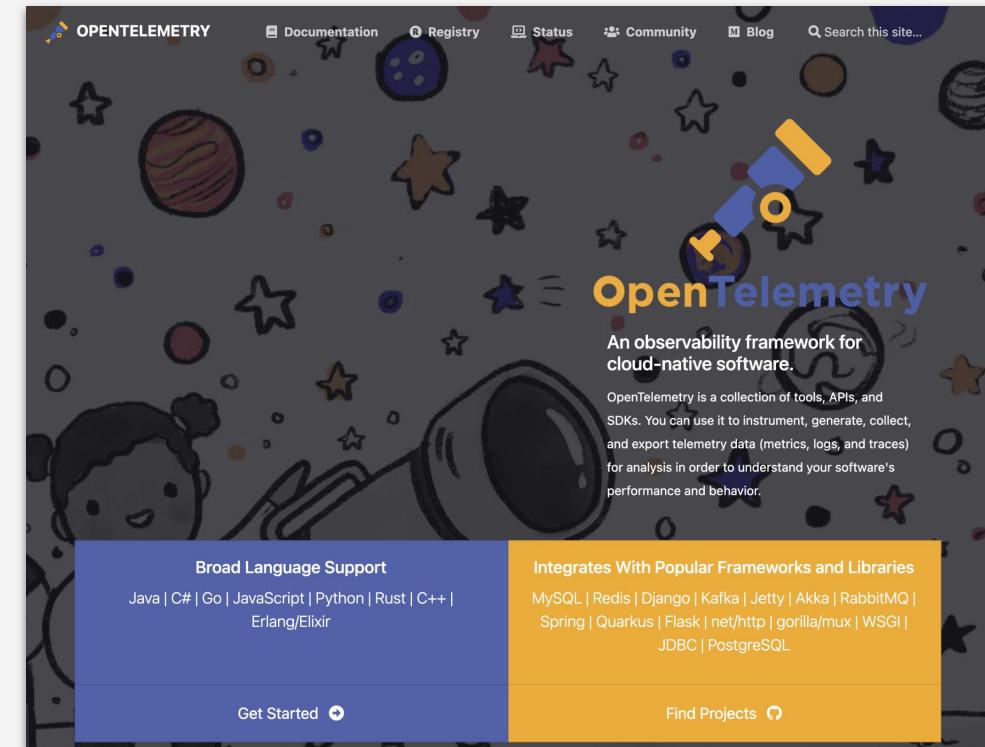


COMMUNITY

Instrumenting Jenkins with OpenTelemetry

What is OpenTelemetry

- CNCF project
- Standardisation of observability and monitoring
 - Semantic conventions (ie. common schema)
 - APIs
 - Protocol
 - Implementations: SDKs, agents and collector
- Unification
 - All layers: infrastructure, application, and user experience
 - All signals: traces, metrics, and logs



Mapping Jenkins concept to Traces

Jenkins status-00004 - Jenkins / Grpc Demo #1

Branch: master Commit: 9da17cc No changes

Pipeline: Checkout → Build Custom Environment → Build → Test → Prepare Docker Image → Build and Push Docker Image → Deploy to DEV → Verify Deployment → Deploy to LIVE

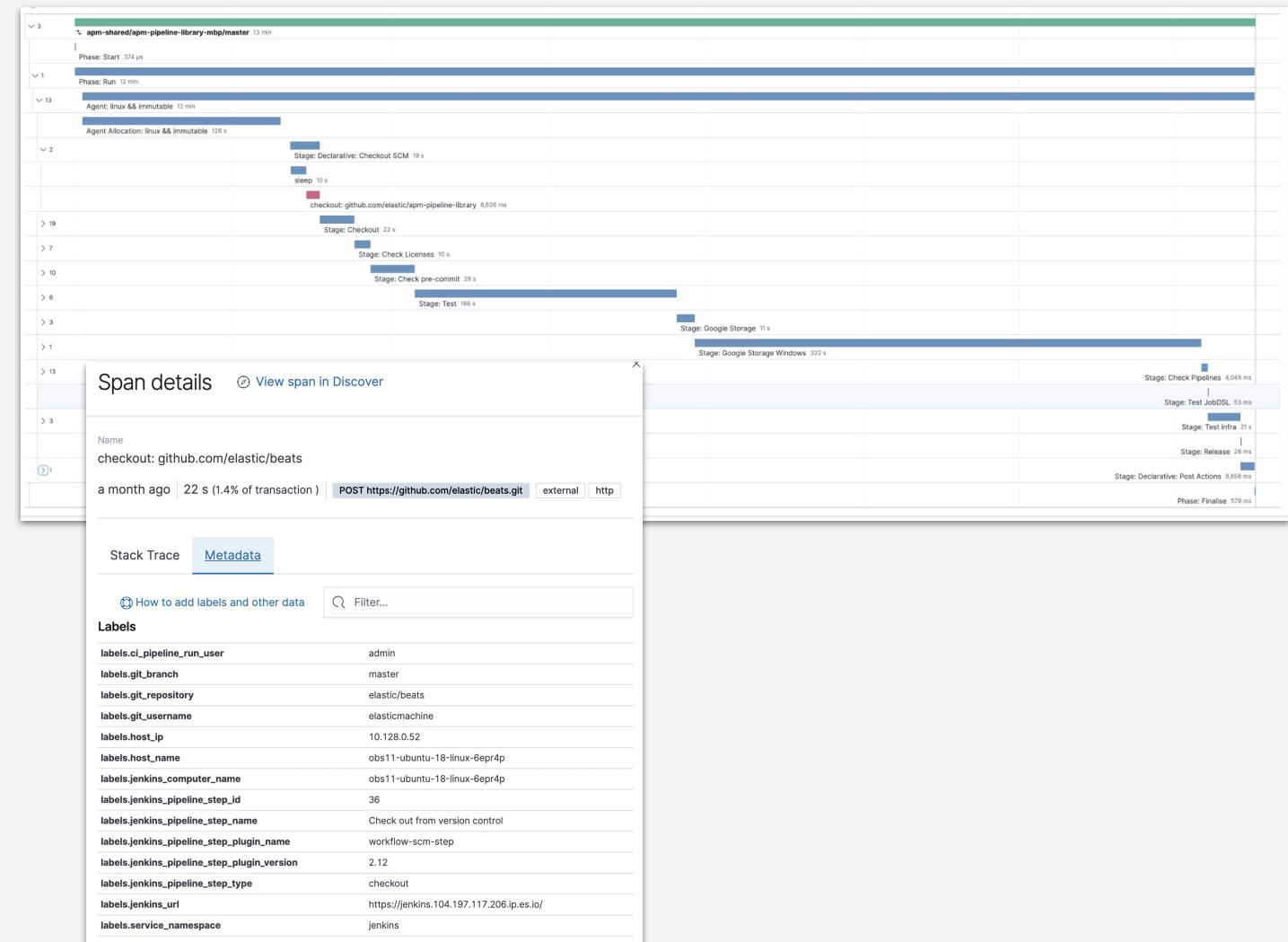
Console Output:

```
Started by user Trans_Ann_Git (OPSI)
Loading library phoenix/phoenix-sharedpipeline@master
Setting origin to ssh://git@git.barcapital.com:7999/lb_phoenix/phoenix-sharedpipeline-devops.git
Fetching origin...
�从远程仓库拉取
从 https://git.barcapital.com:7999/lb_phoenix/phoenix-sharedpipeline-devops.git 取得 133 个新文件
Checking out Revision e314b93f3216e83e575d8fecc43cda87b28733 (master)
[Pipeline] stage
[Pipeline] {
[Pipeline] }
[Pipeline] node
Running on master in /app/phoenix/Jenkins2home/workspace/phoenix-Jenkins-test
[Pipeline] {
[Pipeline] }
[Pipeline] httpRequest
HTTP Request: GET
URL: http://192.168.56.101:8080/jenkins/api/json?tree=jobs[0].name,items[name,script]

```

Steps - Deploy to LIVE:

- > Wait for interactive input: 9 hours 25 minutes 36 seconds
- > Shell Script: 0 seconds
- > Shell Script: 0 seconds
- > Shell Script: 1 seconds
 - [Output]: [grpc_Demo_master-NWYJ7ARG6IHZBLYJKMZD6I27AIHDXZ2LXQCPBCI6W3POFL6JSGQO] Running shell script
2 + kubectl --kubeconfig=*** --context=LIVE rollout status deployment/grpc-demo
3 Waiting for rollout to finish: 0 of 1 updated replicas are available...
4 deployment "grpc-demo" successfully rolled out
- > Shell Script: 0 seconds

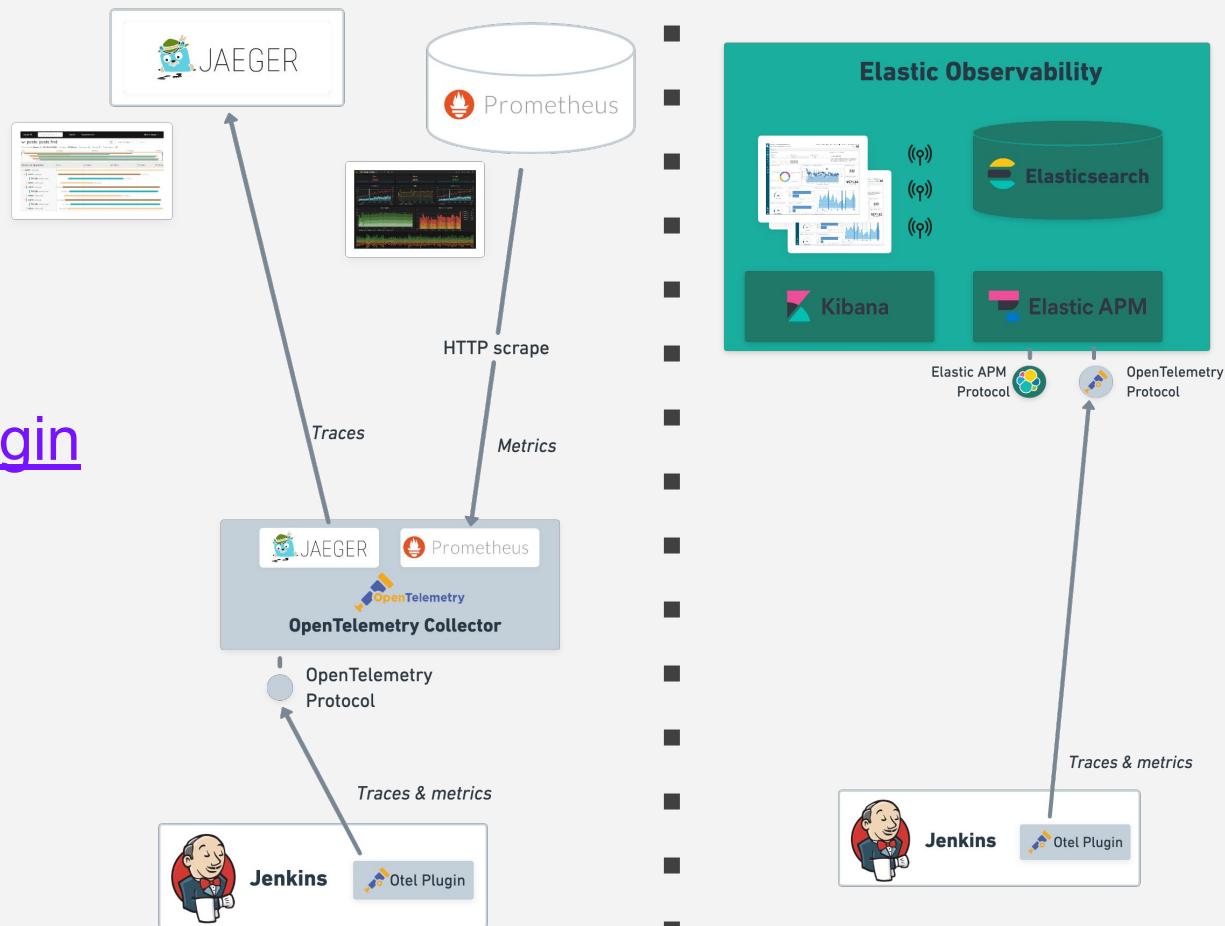


COMMUNITY

Getting Started

Getting started

- Choose your observability backend
 - Elastic, Jaeger...
- Configure the [Jenkins OpenTelemetry Plugin](#)
- Done!



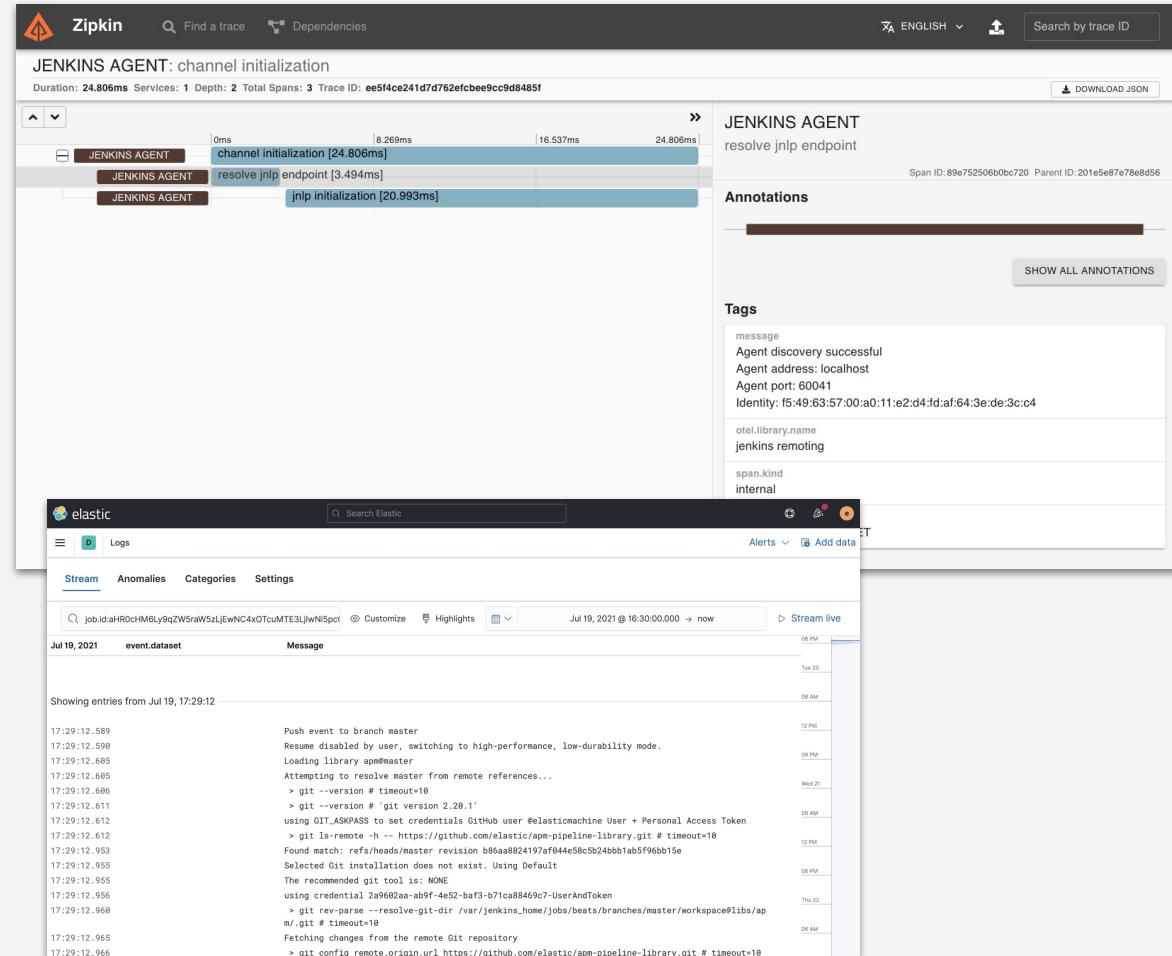
COMMUNITY

What's next

Better Jenkins observability for admin and for developers
Observability for CI/CD, beyond Jenkins

Better Jenkins observability for admins and for developers

- GSOC project to instrument Jenkins Remoting by Akihiro Kiuchi
- Collect logs, including correlation with pipeline traces
- Verify the usability of observability solutions for CI/CD use cases
- Standardising observability for CI/CD introducing semantic conventions for CI/CD



Observability for Infrastructure as Code



- Instrument Ansible / Terraform orchestrations
- Personas: DevOps engineer
- Benefits
 - Investigate transversally on Infra as Code orchestrations
 - Single pane of glass unifying CI/CD pipelines with Infra as Code deployments



Observability for Automated Tests

- Instrument the JUnit of the world
- Personas: DevOps engineer & developers
- Benefits
 - Investigate transversally on tests
 - Single pane of glass unifying CI/CD pipelines with test
 - Optimize CI pipelines



COMMUNITY

Summary

DevOps World sessions on Observability

- **Who Observes the Watchers? An Observability Journey**
 - Breakout session by Victor Martinez Rubio and Ivan Fernandez Calvo
- **Observing the CI/CD in practice**
 - Lighting talk by Victor Martinez Rubio and Ivan Fernandez Calvo

Summary

- Yet Another Jenkins Monitoring Plugin?
 - CI/CD outages are hard to solve
 - Status of running Jenkins in production
 - Using observability for CI/CD
- Instrumenting Jenkins with OpenTelemetry
- Getting started with the Jenkins and OpenTelemetry
- What's next
 - Better Jenkins observability for admins and for developers
 - Observability for CI/CD, beyond Jenkins



Thank you!

Cyrille Le Clerc