keptn

Disclaimer

Paul is **not** a part of the Keptn project, nor has he ever used it in a production system.

This is a "***first look***" presentation.

He is simply interested in improving reliability with **Continuous Delivery**.

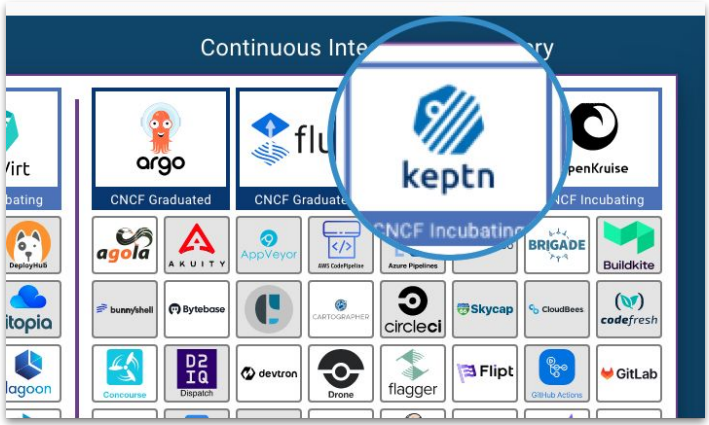
Forgive me...I ran out of time.

keptn

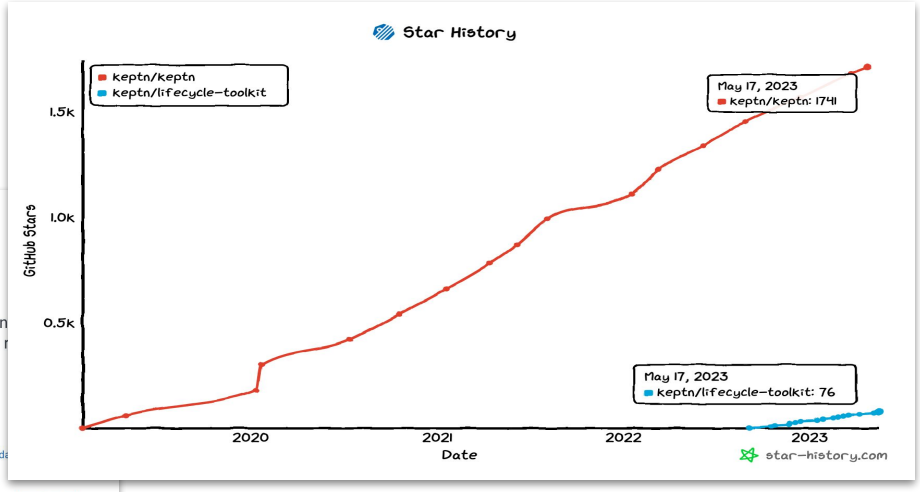
Cloud-native application life-cycle orchestration.

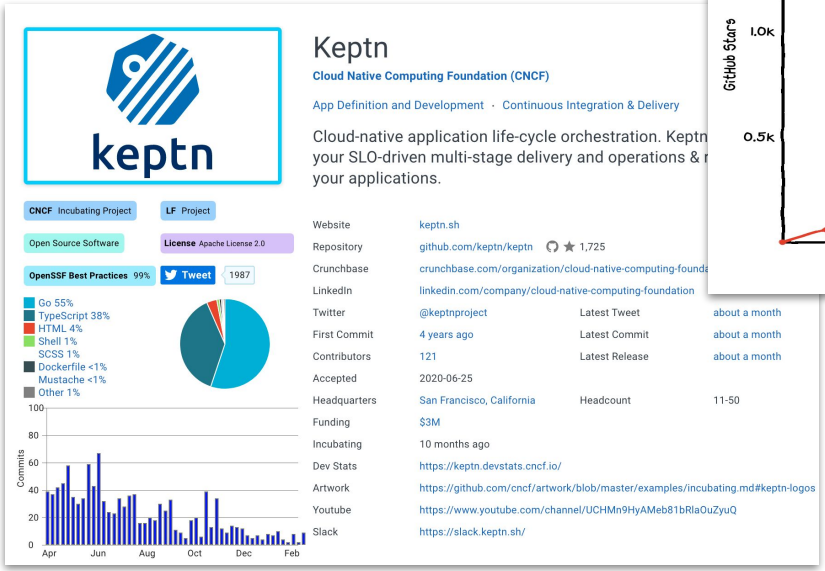
Automates your **SLO-driven**, multi-stage **delivery** and **operations** & **remediation** of your applications.

- CNCF Incubating Project (as of July '22)
- Utilizes Observability Platform
- Based on GitOps
- Declarative & Extensible
- New Lifecycle Toolkit



Project Activity





@keptnProject

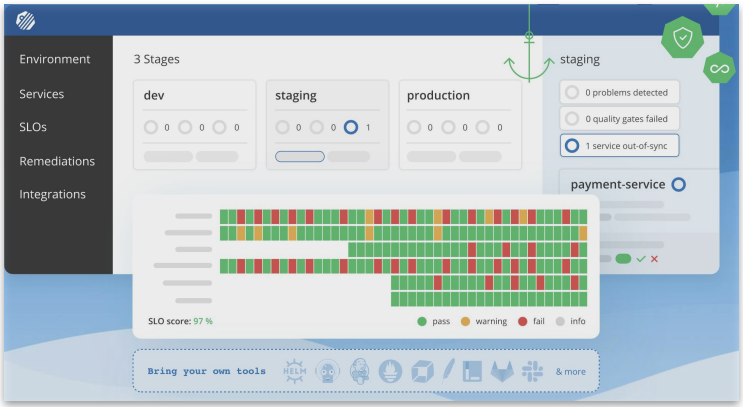
https://github.com/keptn/keptn
https://github.com/keptn/lifecycle-toolkit

Keptn v1

The “OG”

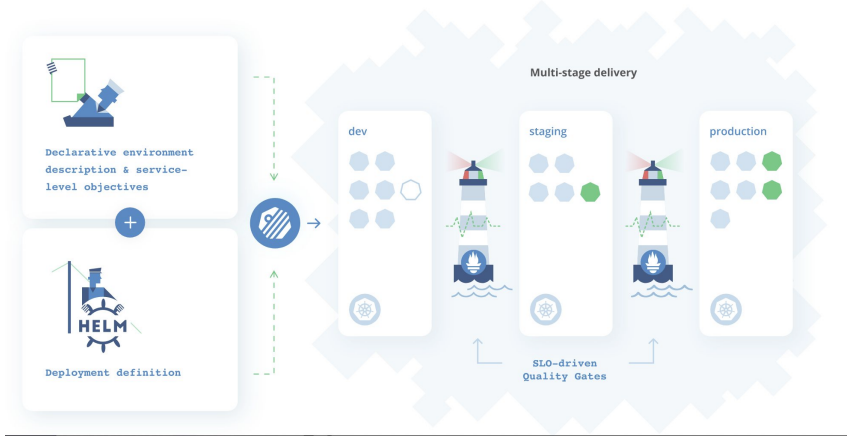
Keptn v1

- Dashboard with **deployment status**
- Automated **remediation** based on SLOs
- Event-driven using **CloudEvents**
- **GitOps** mentality



Keptn v1

- Deployment Pipeline with **Stages**
- Defined within `shipyard.yaml`
- **Lighthouse** service to evaluate results
- Configurable **alerts**
- **Quality Gates**



keptn

Keptn v1

No longer the direction of the project.

keptn

And **that's all** I have to say about that.

- *Forrest Gump*

keptn

Lifecycle Toolkit

Cloud Native

keptn

Lifecycle Toolkit

keptn

Lifecycle Toolkit

- Going forward, **this is the way**.
- **Desire** to be Kubernetes- and Cloud-Native
- Uses Labels & Annotations to **intercept** deployments going to scheduler
- Quality Gates are now **Evaluations**

keptn

Lifecycle Toolkit

Three primary use cases:

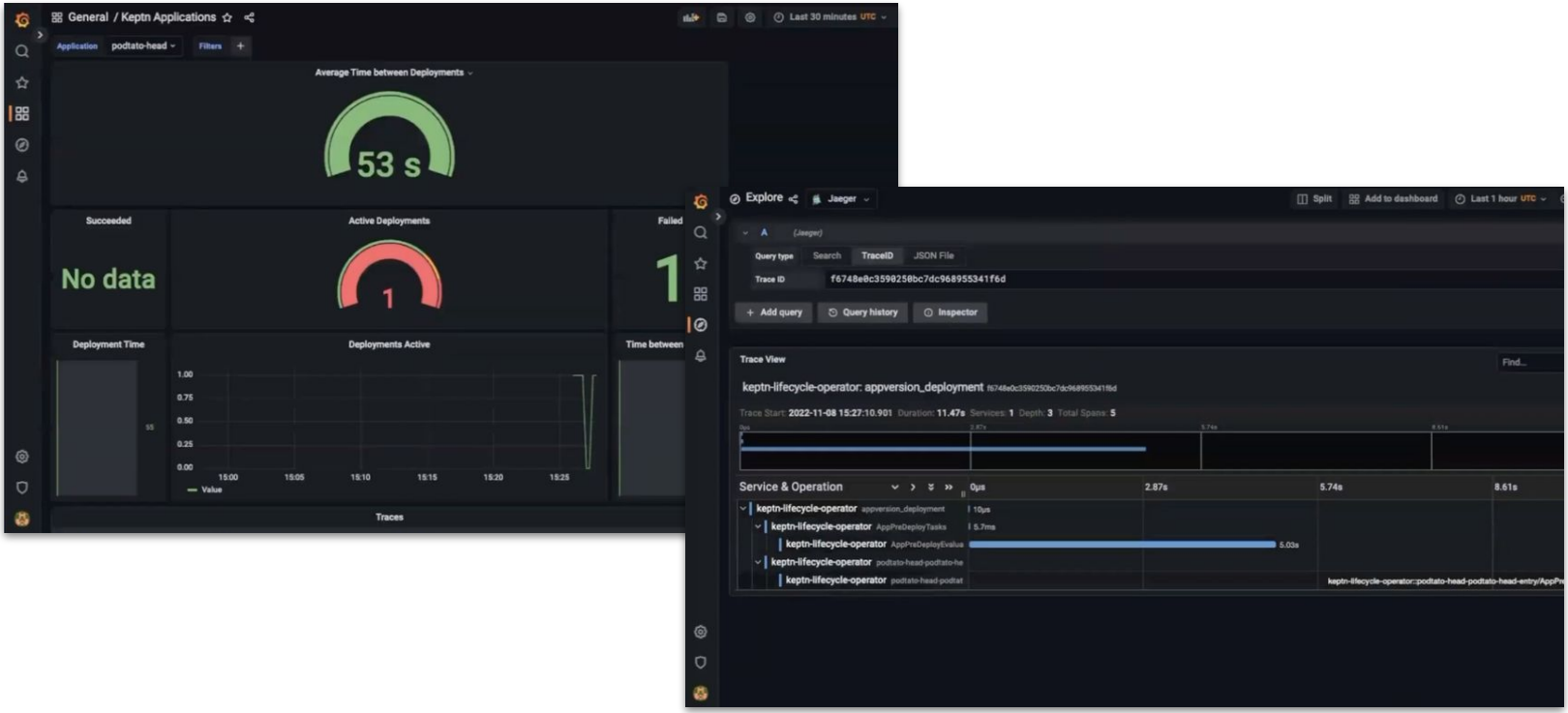
- Make any deployment **observable**
- Standardized **access** to observability data
- **Orchestration** of deployment checks

keptn

Make any deployment observable

- Application-aware DORA **metrics**
  - Deployment Frequency (DF), Lead Time for Changes (LT), Mean Time to Recovery (MTTR), Change Failure Rate (CFR)
- **Troubleshoot** failed deployments
- Trace deployments from **Git** to **Cloud**
- Bring your own tool: ArgoCD, Flux, GitLab, etc.

Dashboards and Traces



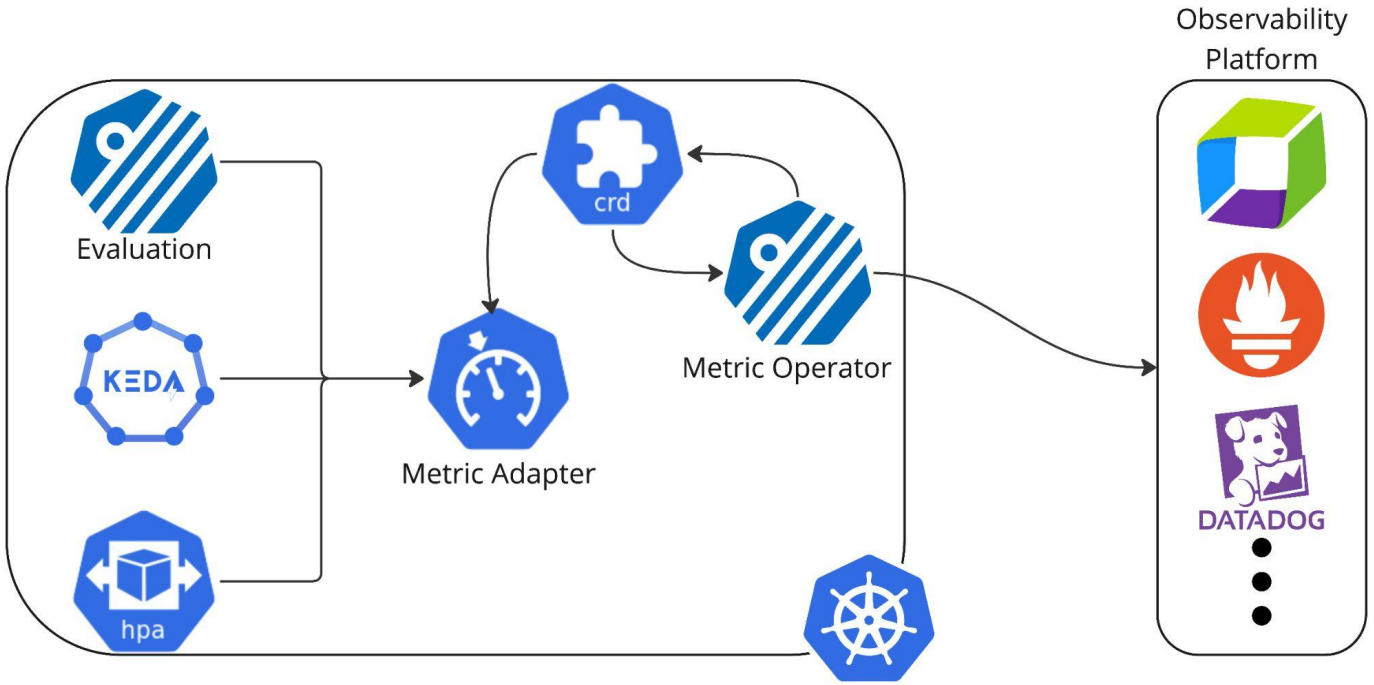
keptn

Standardized access to observability data

- Define metrics **once** for your datastore
  - Prometheus, AWS, Azure, GCP, Dynatrace, DataDog
- **Access** those metrics via Metrics Server or Kubernetes Metrics API
- **Eliminate** need for multiple plugins for Argo Rollouts, KEDA, HPA, etc.

keptn

Metrics Server

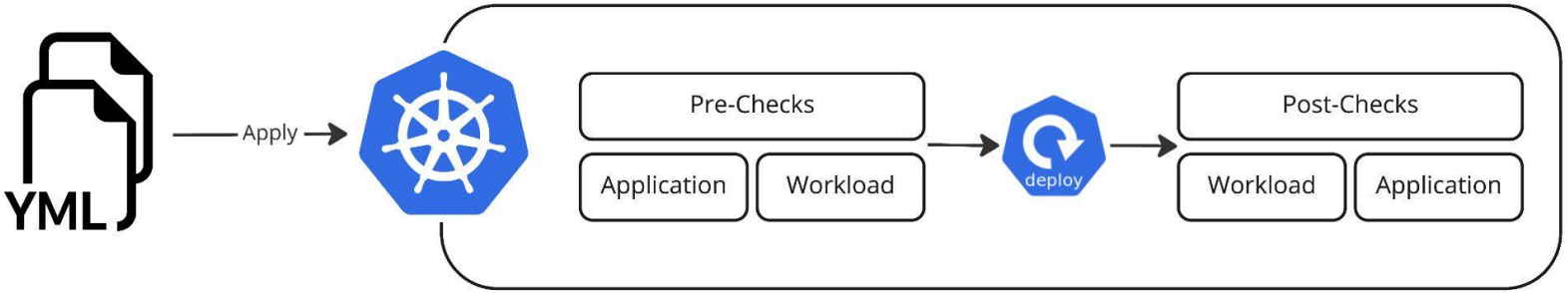


keptn

Orchestration of deployment checks

- Custom **pre**-deployment checks
  - Validate external dependencies
  - Confirm clean security scans
- Custom **post**-deployment checks
  - Execute load tests
  - Send notifications to stakeholders
- Automatic **validation** against Service Level Objectives (SLOs)

Basic Workflow



keptn

Demo

THIS SLIDE INTENTIONALLY LEFT BLANK

keptn

What I couldn’t show you

- Grafana **dashboards** for Application deployments
- **Traces** for deployments in Jaeger or Tempo
- How to **connect metrics** to the Grafana Cloud
- Trigger a **k6 test** as a “quality gate”

- probably **much more…**

keptn

Project Roadmap

- Multi-stage and Cluster observability
- *Instances* as first-class types
- Introduce concept of Promotion
- Native container support for Tasks
- Propagate result of Task execution
- and **much more...**

keptn

Final Thoughts

My feelings after this initial experimentation.

keptn

Evaluation

- Great potential
- Confusing message for project
- Documentation gaps
- More examples please
- Reevaluate in 6-12 months
- Will I contribute?... **YES** (Will you?)

** Based upon Paul's very subjective scoring.*

keptn

Thank you!

@javaducky

https://github.com/javaducky/demo-keptn