KubeCon | CloudNativeCon

Europe 2020

Virtual

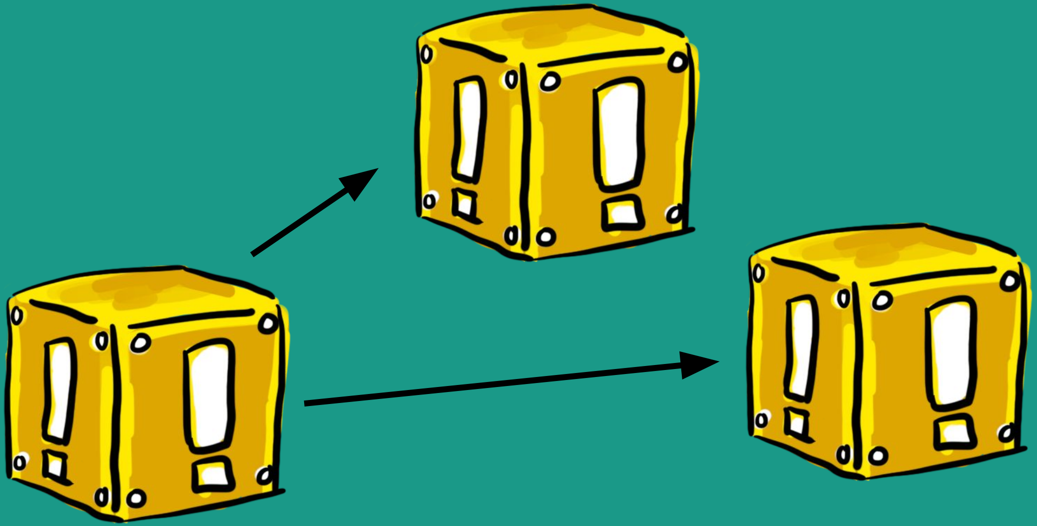
Mario meets the Robocat: Lessons from Dogfooding Tekton

Dibyo Mukherjee, @_dibyom, Google
Andrea Frittoli, @blackhip76, IBM

Agenda

- What is Tekton ?
- Mario’s Plumbing
- Tekton CI
- Demo
- Future Work

What is Tekton ?



Tekton

Tekton is a powerful and flexible open-source framework for creating CI/CD systems, allowing developers to build, test, and deploy across cloud providers and on-premise systems.

Governed by the Continuous Delivery Foundation

Contributions from Google, Red Hat, Cloudbees, IBM, Pivotal and many more

Tekton Goals

Standardization

Tekton standardizes CI/CD tooling and processes across vendors, languages, and deployment environments. It works well with Jenkins, Jenkins X, Skaffold, Knative, and many other popular CI/CD tools.

Built In Best Practices

Tekton lets you create CI/CD systems quickly, giving you scalable, serverless, cloud native execution out of the box.

Maximum Flexibility

Tekton abstracts the underlying implementation so that you can choose the build, test, and deploy workflow based on your team’s requirements.

Tekton Projects

Building Blocks

Pipelines: standardized definitions for building CI/CD pipelines such as Tasks, Pipelines, and Steps
Triggers: Run pipelines from webhook events.

Reusable CI/CD Definitions

Catalog and *Hub* host a variety of reusable task definitions.

Tooling

Dashboard - a web UI for Tekton
tkn - a *CLI* for interacting with Tekton
Operator - manage Tekton installation and upgrades.

Experiments

DSLs, Results storage and querying, Image formats, supply chain security, Git and polling operators etc.

Tekton Pipelines

Step

Run commands in a container with volumes, env vars, etc

Task

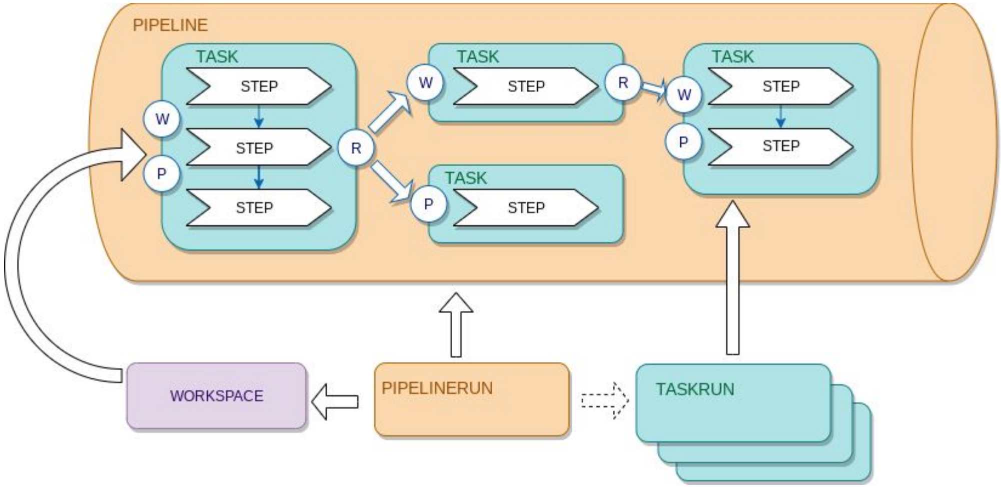
A list of steps that run sequentially in the same pod

Pipeline

A graph of tasks with inputs and outputs executed in a certain order

TaskRun and PipelineRun

An invocation of a task or pipeline with inputs and outputs



Tekton Triggers

EventListener

Listen and process incoming events

Interceptors

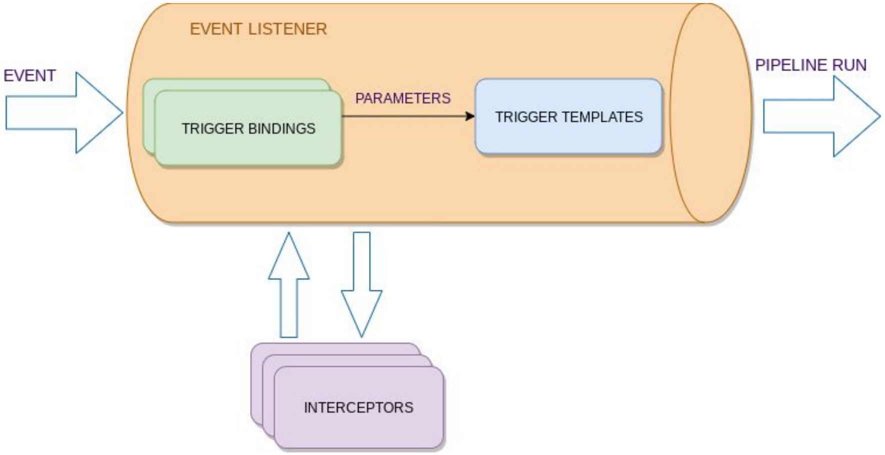
Filter and validate incoming events

Trigger Binding

Extract params from incoming events

Trigger Template

An invocation of a pipeline with inputs and outputs



Mario’s Plumbing

Plumbing

… « the system of pipes, tanks, fittings, and other apparatus required for the water supply, heating, and sanitation in a building. »

For us this means, all tools and configuration files for the testing and automation needs of Tekton:

- Continuous integration system
- release setup
- test infrastructure
- scripts (for the CI, tests, release, infrastructure)
- GitHub issues and pull-request management (labels, /lgtm, …)
- …

Dogfooding: The Beginning

In the beginning,

- Use Prow, a k8s based CI/CD tool used by Kubernetes, Knative, etc.
- Share scripts and containers with Knative
- Use infrastructure from Google Cloud

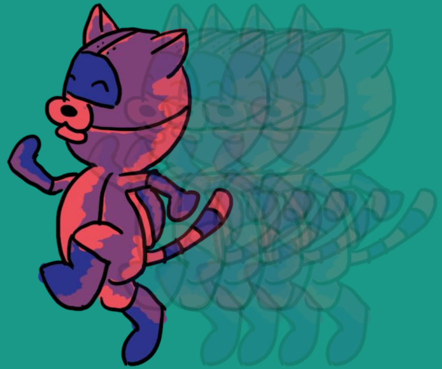
First steps in Dogfooding

- Release Tekton using Tekton
- Setup dedicated cluster for experimentation
- Integrate Tekton Pipelines as a Prow agent

Meeting the Robocat

- Nightly releases for Tekton projects
- Nightly builds for infra containers
- Continuous deployment of Tekton releases, configuration, and other resources
- Automated GitHub org management using Peribolos and Triggers

Tekton CI testing Tekton CI testing Tekton CI testing...



Tekton based CI

Overlays and bindings

- Common interface to Pipelines

Conditions

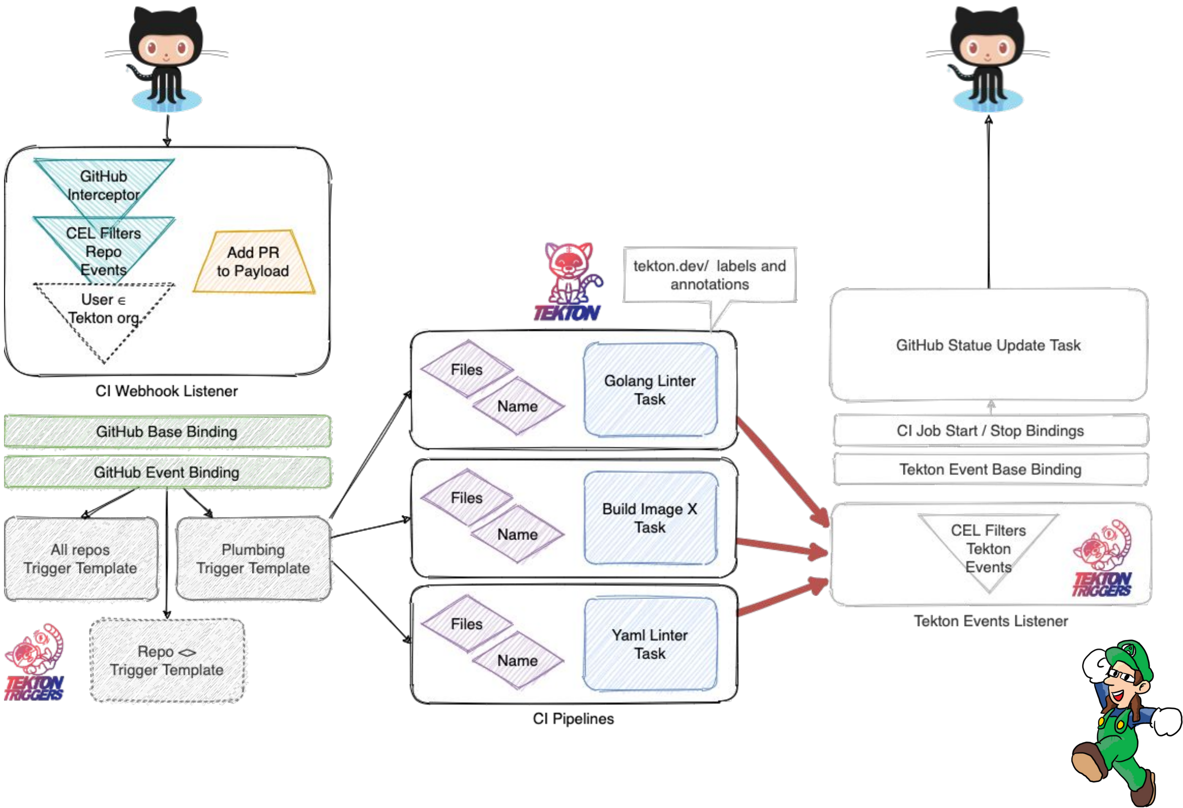
- Skip unnecessary CI Jobs

Tasks

- Run CI Jobs

Cloud Events

- Trigger Sync back to GitHub



GitHub Interceptor

CEL Filters Repo Events

User ∊ Tekton org

Add PR to Payload

CI Webhook Listener

GitHub Base Binding

GitHub Event Binding

All repos Trigger Template

Plumbing Trigger Template

Repo <> Trigger Template

tekton.dev/ labels and annotations

Files

Name

Golang Linter Task

Files

Name

Build Image X Task

Files

Name

Yaml Linter Task

CI Pipelines

GitHub Statue Update Task

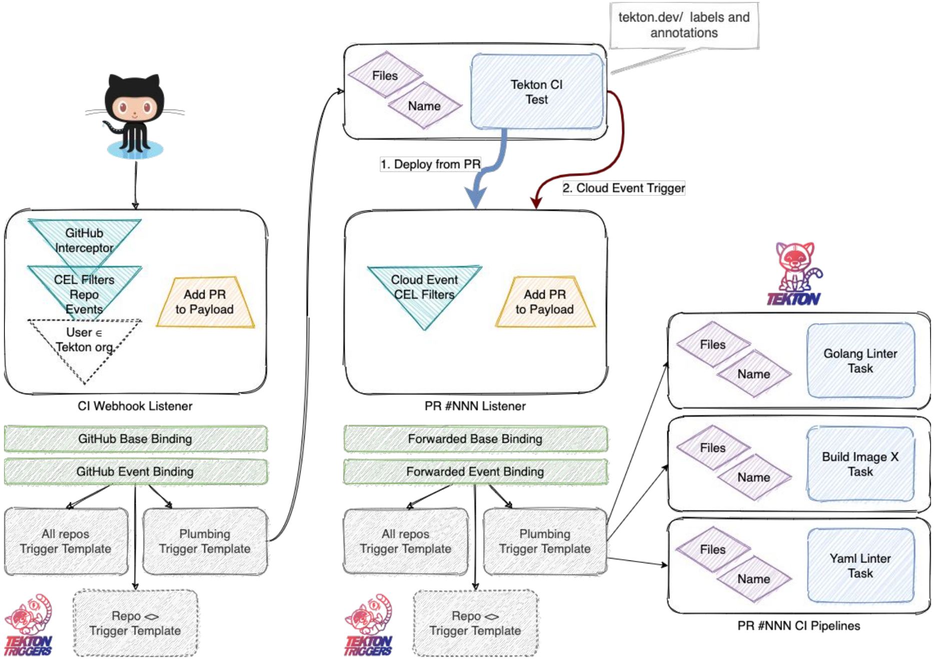
CI Job Start / Stop Bindings

Tekton Event Base Binding

CEL Filters Tekton Events

Tekton Events Listener

Testing the test pipelines



tekton.dev/ labels and annotations

Files

Name

Tekton CI Test

1. Deploy from PR

2. Cloud Event Trigger

GitHub Interceptor

CEL Filters Repo Events

User ∊ Tekton org

Add PR to Payload

CI Webhook Listener

GitHub Base Binding

GitHub Event Binding

All repos Trigger Template

Plumbing Trigger Template

Repo <> Trigger Template

Cloud Event CEL Filters

Add PR to Payload

PR #NNN Listener

Forwarded Base Binding

Forwarded Event Binding

All repos Trigger Template

Plumbing Trigger Template

Repo <> Trigger Template

Files

Name

Golang Linter Task

Files

Name

Build Image X Task

Files

Name

Yaml Linter Task

PR #NNN CI Pipelines

Testing CI Changes:

- Trial and error? No!

Motivation:

- No access to the CI Cluster
- Reduce need for own infra
- Troubleshooting

Solution:

- Deploy isolated CI from **trusted** PR
- Forward GitHub Event as Cloud Event
- Annotations + Trigger Bindings

Demo

Fixing our pipes

a.k.a Future Work

So, What did we Learn?

We made all the mistakes!

- Verify your inputs
- Don't override your last release
- Automation helps here

There is a lot of YAML

- Lots of boilerplate
- Kustomize helps somewhat
- Maybe, DSLs in the future?

Testing changes to CI setup

- Local testing is hard to do
- Setting up pipelines

Debuggability/Tracing

- Using Labels + standard k8s tooling
- UI Integration would be nice

Automate infra setup:

- Refresh cluster after security incident
- Manual changes make this hard

Not everything needs to be in a Pod

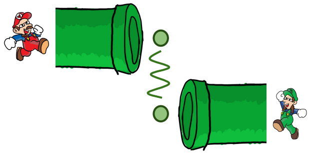
- Terraform, Ansible for seeding infra
- Custom Tasks

Document, document, document!

Pipes we're Building

Using Task Results in Finally

-> *TEP#0004*

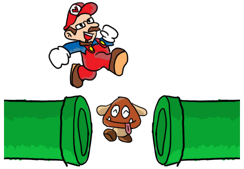


Notifications

-> *Cloud Events Based.*

Skip / Simple Conditionals

-> *TEP#0007*

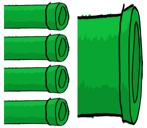


Metrics, Tracing, Debugging

-> *TEP#NNNN*

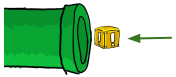
Task Hooks, Switch and Loops

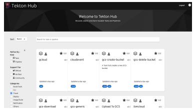
-> *Under discussion*



Tekton OCI Bundle

-> *TEP#0005*





...and more!

Custom Tasks

-> *TEP#0002*

Trigger CRD

-> *TEP#NNNN*

Event Listener as Knative Service

-> *TEP#008*

Ephemeral Credentials

-> *and GitHub App!*

Logs and Test Results:

- Log parsing
- Test result dashboard, flake analysis

On CI itself:

- Use more Tasks from the Catalog
- “Depends-on” for cross-repo testing
- E2E Tests as Tekton pipelines
- Cloud neutral tests
- Monitoring of CI Services
- /meow!

Come and Join Us

We're on tektoncd.slack.com!

To Join: https://bit.ly/2D2vDqh

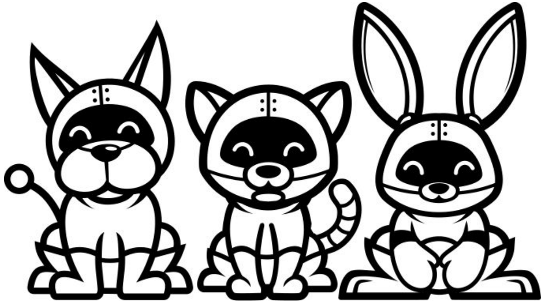
Thank you !

Image credits: Christie Wilson (@bobcatwilson)

Links & References

- Tekton: https://tekton.dev
- Tekton on GitHub: https://github.com/tektoncd
- Tekton Community: https://github.com/tektoncd/community
- Tekton Friends: https://github.com/tektoncd/friends
- CDF: https://cd.foundation/, https://github.com/cdfoundation

- Plumbing Repo: https://github.com/tektoncd/plumbing
- K8s Test-Infra: https://github.com/kubernetes/test-infra/



KubeCon | CloudNativeCon

Europe 2020

Virtual

KEEP CLOUD NATIVE CONNECTED