



Scaling Pipelines with Tekton

Andrea Frittoli
Developer Advocate
andrea.frittoli@uk.ibm.com



Scaling Tekton



Introduction
Authoring
Running
Bottlenecks
Q&A



Introduction



Photo by Mike Benna, CC0



Tekton is an open-source framework for creating CI/CD systems

Cloud Native
Serverless,
Scalable
Pipelines



Standardization
Built In Best
Practices
Maximum
Flexibility

Core Projects
➤ Pipeline
➤ Triggers
Tooling:
➤ CLI *tkn*
➤ Dashboard
➤ Operator

Discovery:
➤ Catalog
➤ Hub
Adds-on:
➤ Results
➤ Chains
➤ Experiments

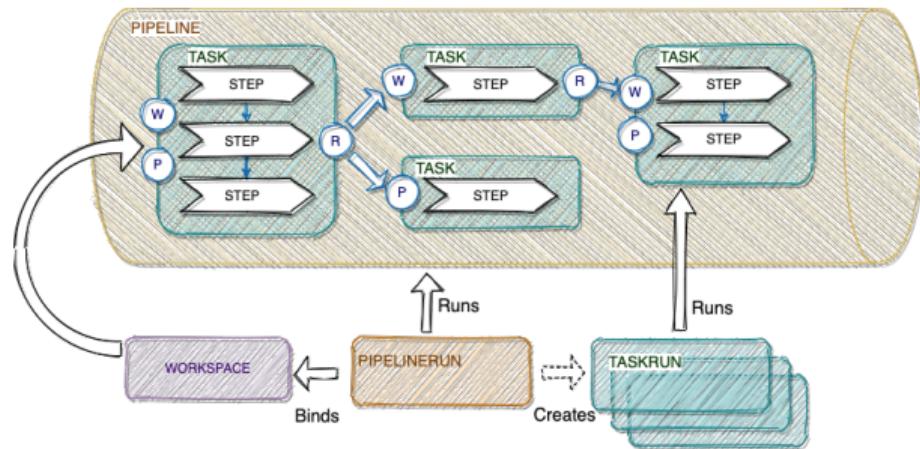


A bit of history

- › From Knative Built, to Pipeline
- › Extend the k8s API with CRDs
- › Tekton and the CDF



- › Definitions: Step, Task, Pipeline
- › Bindings: Workspaces, Parameters, Results
- › Execution: TaskRun, PipelineRun



Authoring



Building Blocks



Steps:

- Off the Shelves containers
- Small scripts

Tasks:

- Solve one specific problem
- Owned by a team
- Distributed maintenance

Reusability:

- Discovery
- Versioning
- Best Practices



Catalog and Hub

Discovery:

- › Kubernetes cluster
- › Tekton Catalog
- › Tekton Hub & API

Versioning:

- › Tekton Bundles
- › tkn bundle [push|list]
- › gcr.io/tekton-releases/catalog/upstream/

The screenshot shows the Tekton Hub interface. At the top, there's a search bar with 'build' and a 'Login' button. Below it, a banner says 'Welcome to Tekton Hub' and 'Discover, search and share reusable Tasks and Pipelines'. The main area is a catalog of tasks, with a sidebar for filtering by kind (Task) and category (Tekton). The tasks listed are:

- buildid** v0.1: Given a base version, this task generates a unique build id by appending the base-version to the current timestamp. Updated 3 months ago.
- buildah** v0.2: Buildah task builds source into a container image and then pushes it to a container registry. Updated 3 months ago.
- buildkit** v0.1: This Task builds source into a container image using Moby BuildKit. Updated 3 months ago.
- buildpacks** v0.2: The Buildpacks task builds source into a container image and pushes it. Updated 3 months ago.
- buildpacks-photon** v0.1: The Buildpacks Photon task builds source into a container image. Updated 3 months ago.
- golang build** v0.1: This Task is Golang task to build Go projects. Updated 3 months ago.

On the right side of the screen, a detailed view of the 'golang build' task is shown, displaying its YAML configuration:

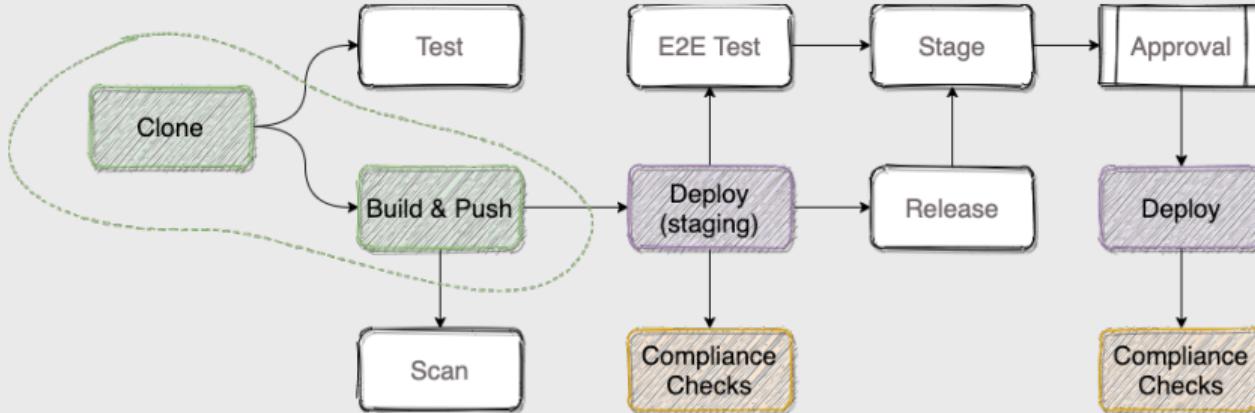
```
apiVersion: v1
kind: Task
metadata:
  name: "git-cl"
  catalog: "0.0"
  type: "Task"
  owner: "testers"
  type: "community"
spec:
  inputs:
    - id: "34"
      name: "git"
      type: "Git"
      description: "This task can be used to perform git operations.\n`git` command that needs to be run can be\n`git pull` or `git push`."
      url: "https://raw.githubusercontent.com/tektoncd/catalog/master/task/git-cl/0.1/git-cl.yaml"
      updatedAt: "2020-10-26T00:01:20Z"
  outputs:
    - id: "34"
      name: "git-cl"
      type: "Git"
  steps:
    - id: "34"
      name: "git-cl"
      type: "Task"
      description: "This task can be used to perform git operations.\n`git` command that needs to be run can be\n`git pull` or `git push`."
      url: "https://raw.githubusercontent.com/tektoncd/catalog/master/task/git-cl/0.1/git-cl.yaml"
      updatedAt: "2020-10-26T00:01:20Z"
  tags:
    - id: "34"
      name: "git"
      type: "Task"
    - id: "34"
      name: "git-cl"
      type: "Task"
  labels:
    - id: "40"
      name: "golang-build"
      catalog: "0.0"
      type: "Task"
      owner: "testers"
      type: "community"
      kind: "Task"
```

Best Practices:

- › Params, Results, Workspaces
- › Portability



Pipelines



Short pipelines:

Building blocks

- Conditions
- Optional Inputs

Large Pipelines:
How to:

- Add a Branch
- Swap a Block
- Access Control
- Distribute

Distribute:

- Team
- Access
- Namespace
- Cluster
- Platform

One workflow:

- Pipeline in Pipeline (experimental)
- CloudEvents



Running



Photo by Ray Bilcliff, CC0



Triggers

Run Pipelines on Events:

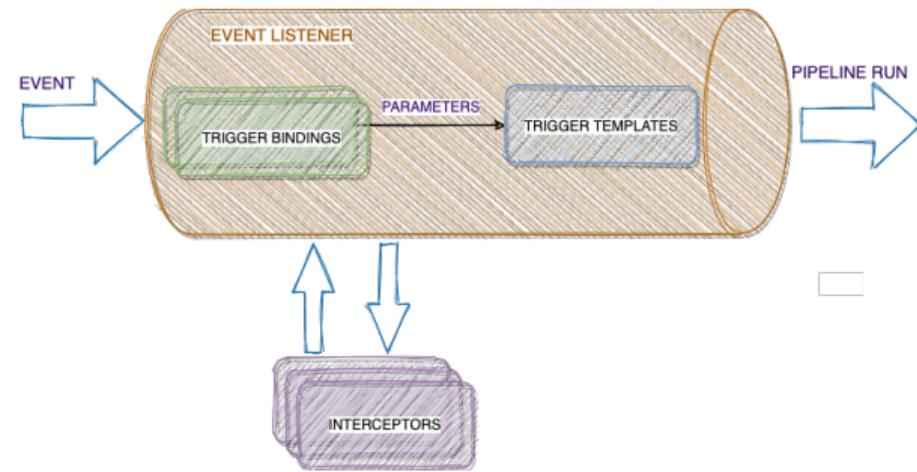
- › HTTP POST
- › Pull Request, Image Published

Components:

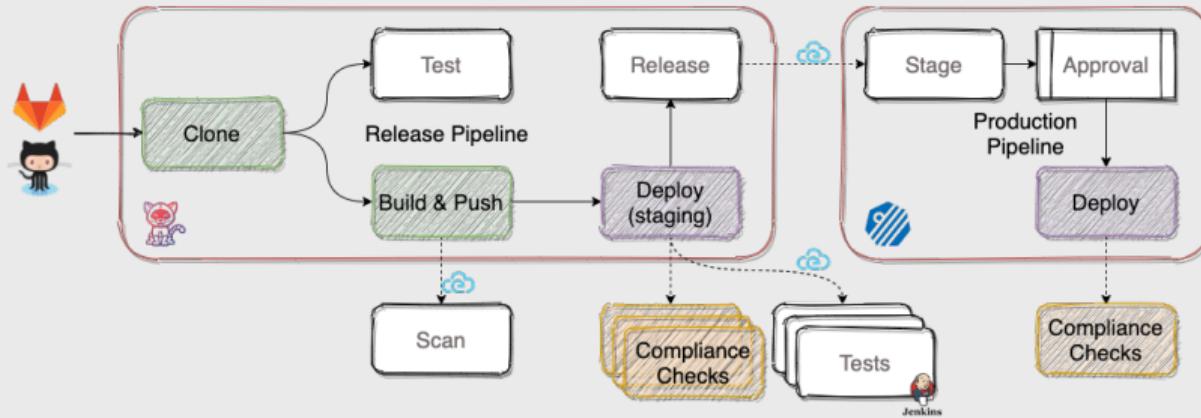
- › Receive: Event Listener
- › Filter: Interceptors
- › Run: Binding and Templates

Generate Events:

- › k8s and CloudEvents
- › Start, Run and Stop



Events



Receiving and sending events

Composite Workflow

Separate Ownership

Dynamically add tasks

Reuse pipeline

Heterogeneous parts

Issues:
Custom integrations
Orchestration
Visibility

Events in CI/CD
Special Interest Group



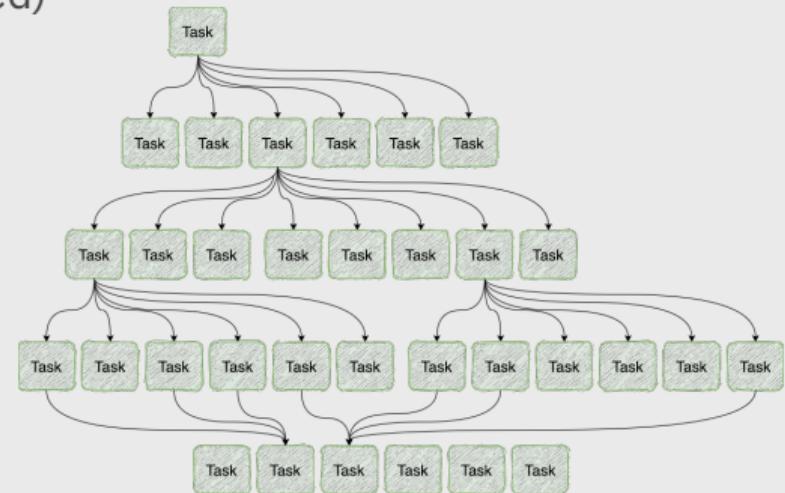


Bottlenecks



Very Large pipelines

- › Directed Acyclic Graphs
- › Large Pipelines (100 nodes, densely connected)
- › Scale Issues?
- › DAG build on every reconcile
- › Hundreds of nodes and connections
- › Optimized code in the DAG computation
- › Maintainability
- › Pod Overhead



Under Pressure - Running **many** Pipelines

Concurrent execution

- Cached K8s API calls: Informers
- Scale Up Controllers: LeaderElection
- Cluster Resources: K8s scheduler

In real life

- Thousands Tasks/month upstream
- Millions containers/month @IBM
- Throttling (for security too)

Potential enhancements:

- Throttling pipeline execution
- Optimize Execution Runtime
- Tekton custom scheduler

Cluster pollution?

- Tekton Results



Data and I/O

Reusability vs. Efficiency

Common Tasks (→Pods):

- › Clone a git repository
- › Build a container image
- › Download content from storage
- › Run tests

Pods scheduled independently

- › Shared storage required (PVC)
- › Extra I/O
- › Extra execution time

Data in Pipelines

Meta: "Results", Large: "Workspaces"

Alternatives:

- › Multiple Tasks in a Pod
- › Re-usable Steps
- › Custom scheduler





Thank You! Questions?

References

- › Come and Join Us at Tekton!
- › Tekton community: github.com/tektoncd/community
- › Slides: github.com/afrittoli/scaling_pipelines_with_tekton
- › Tekton: tekton.dev
- › Tekton on GitHub: github.com/tektoncd
- › Tekton Results: github.com/tektoncd/results
- › Tekton Hub: hub.tekton.dev
- › Performance TEP: [TEP-0036](#)
- › Metrics TEP: [TEP-0006](#)

