

Scaling Pipelines with Tekton

Andrea Frittoli

Developer Advocate

andrea.frittoli@uk.ibm.com

Scaling Continuous Delivery (Virtual)



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

Add-ons:
– Results
– Chain
– Experiments



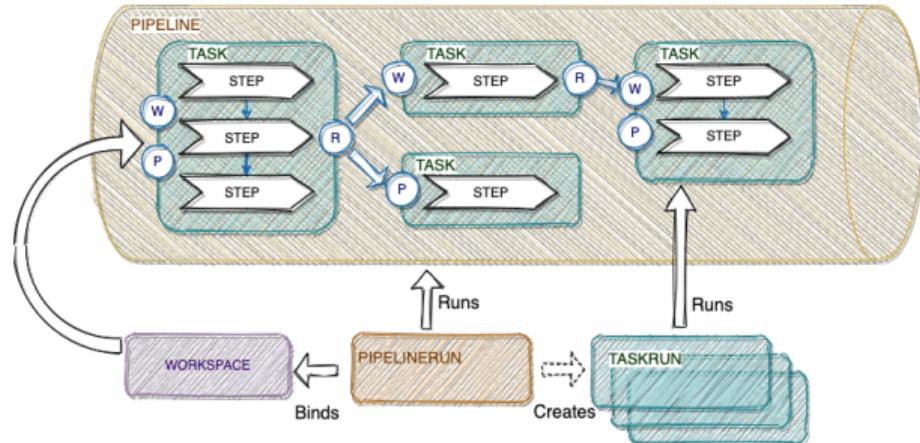
A bit of history

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



CD.FOUNDATION

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



Authoring



Building Blocks



Photo by Antoine Petittvile, CC0

Steps:

- Off the Shelves containers
- Small scripts

Tasks:

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

Reusability:

- Discovery
- Versioning
- Execution efficiency



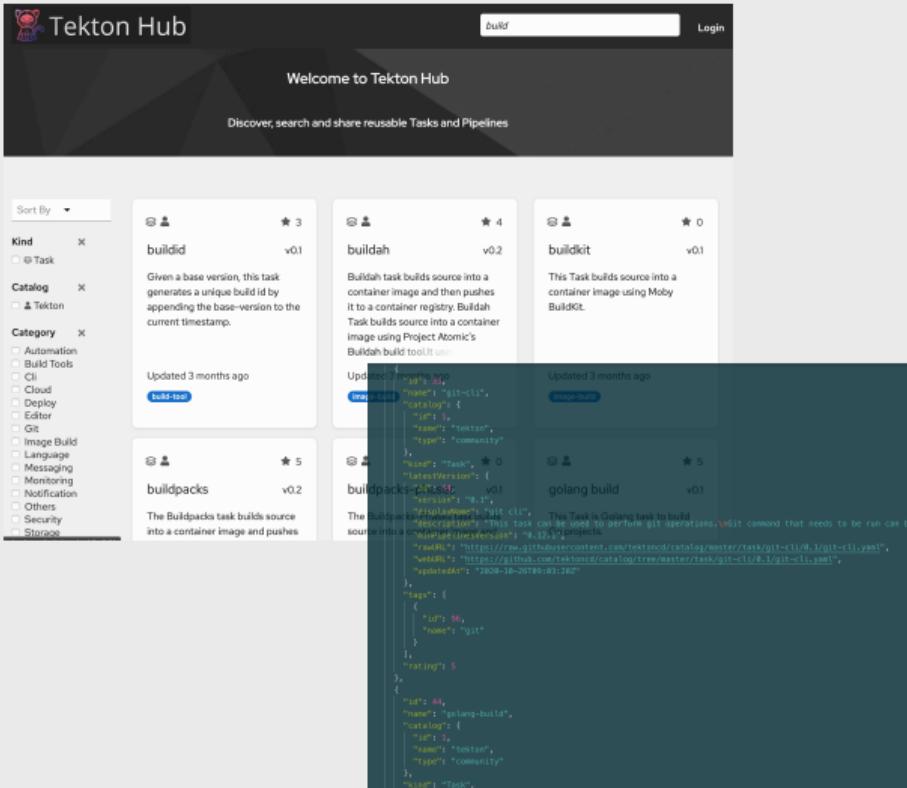
Catalog and Hub

Discovery:

- Kubernetes cluster
 - Tekton Catalog
 - Tekton Hub & API

Improving Reusability:

- Tekton Bundles
 - Hub and Bundles CLI
 - Task Specialization
 - Execution efficiency
 - Multiple Tasks in a Pod
 - Multiple Tasks on a Node



Pipelines

Building block?

Example: clone, build, scan, push, test, release, e2e test, publish

Supported by the catalog.

What scan strategy? Which tests?

I could have multiple test pipelines

Different parts owned by different teams

We still need small pipelines to do one job

How do I connect them?



Running



Photo by Ray Bilcliff, CC0



Triggers

Run Tasks and Pipelines as a reaction to events. Intro to triggers Events can be external (PR, CR, OS) Or Internal -> Pipeline triggers a pipeline



Events

- Receiving and sending events
- Overall workflow made of separate pipelines
- Add branches of pipeline on demand
- Orchestration?
- Interop and event SIG



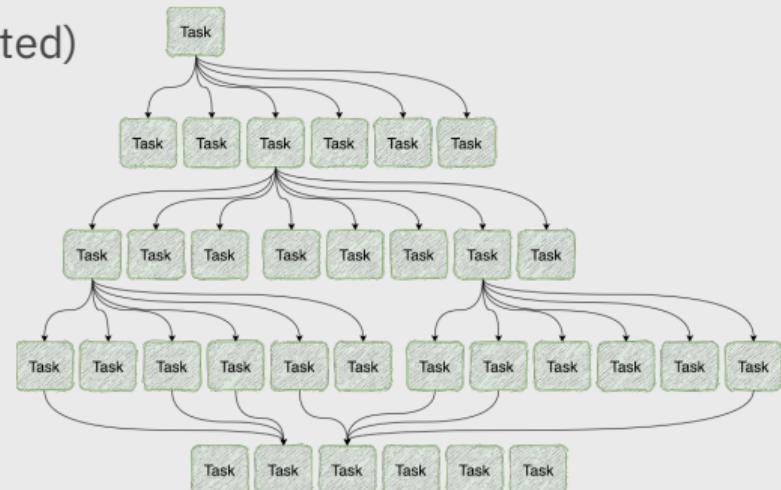


Bottlenecks



Growing pipelines

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



Under Pressure

Concurrent execution

- Cluster Resources: K8s scheduler
- K8s API: Informers
- Tekton Controllers: LeaderElection

In real life

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

Potential enhancements:

- Throttling pipeline execution
- Tekton custom scheduler

Upcoming features:

- Tekton Results
- Performance testing
- Metric improvements



Data and I/O

Reusability vs. Efficiency

- Task == Pod
- Common tasks:
 - 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
 - Multi-region clusters

Data in Pipelines

- Metadata: "Results"
- Large data: workspace
- 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
- Performance TEP: [TEP-0036](#)
- Metrics TEP: [TEP-0006](#)
- Tekton Results: github.com/tektoncd/results

