

# 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



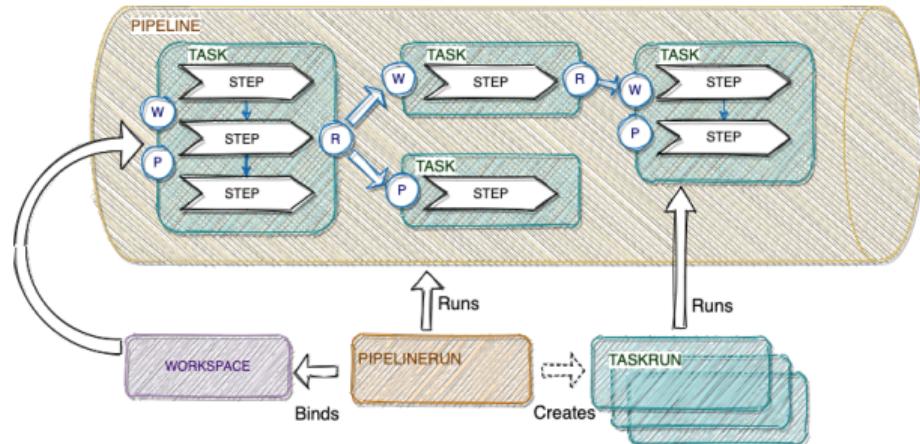
# A bit of history

- 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



# Steps, Tasks and Teams

- units of reuse
- Steps for small scripts of OTS images
- we don't want large shell scripts embedded in YAML
- Tasks == pods, cheap I/O
- Tasks do one thing well, stable interface
- Tasks are re-usable, produced by "experts"



# Catalog and Hub

- Sharing Tasks and discovery
- Hub and API
- Tekton Bundles



# The Pipeline

- Ever growing pipeline
- Multiple teams add to it



# Running



Photo by Ray Bilcliff, CC0



# Pipelines & Triggers

- Intro to triggers
- Running pipelines through triggers



# Events

- Receiving and sending events
- Further spread responsibility, break the pipeline
- Interop and event SIG



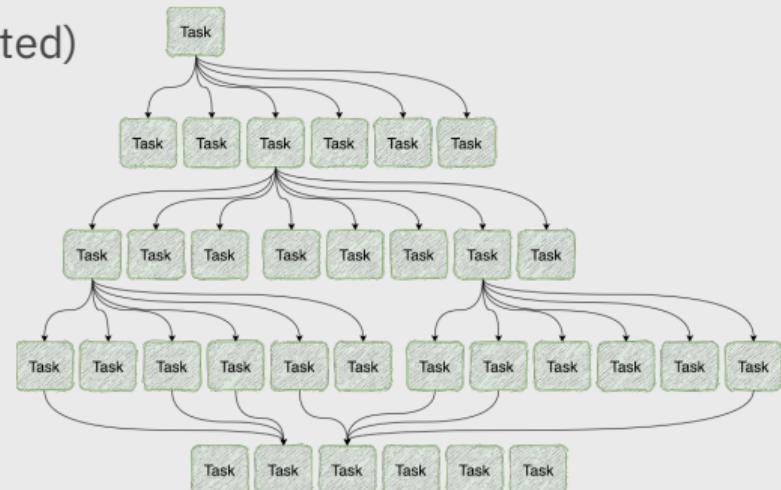


# Bottlenecks



# 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](https://github.com/tektoncd/community)
- Slides: [github.com/afrittoli/scaling\\_pipelines\\_with\\_tekton](https://github.com/afrittoli/scaling_pipelines_with_tekton)
- Tekton: [tekton.dev](https://tekton.dev)
- Tekton on GitHub: [github.com/tektoncd](https://github.com/tektoncd)
- Performance TEP: [TEP-0036](#)
- Metrics TEP: [TEP-0006](#)
- Tekton Results: [github.com/tektoncd/results](https://github.com/tektoncd/results)

