Cloud Native Applications Workshop
Tekton Overview

WW Developer Advocacy

IBM **Developer**

Contents

History of Tekton

What is Tekton?

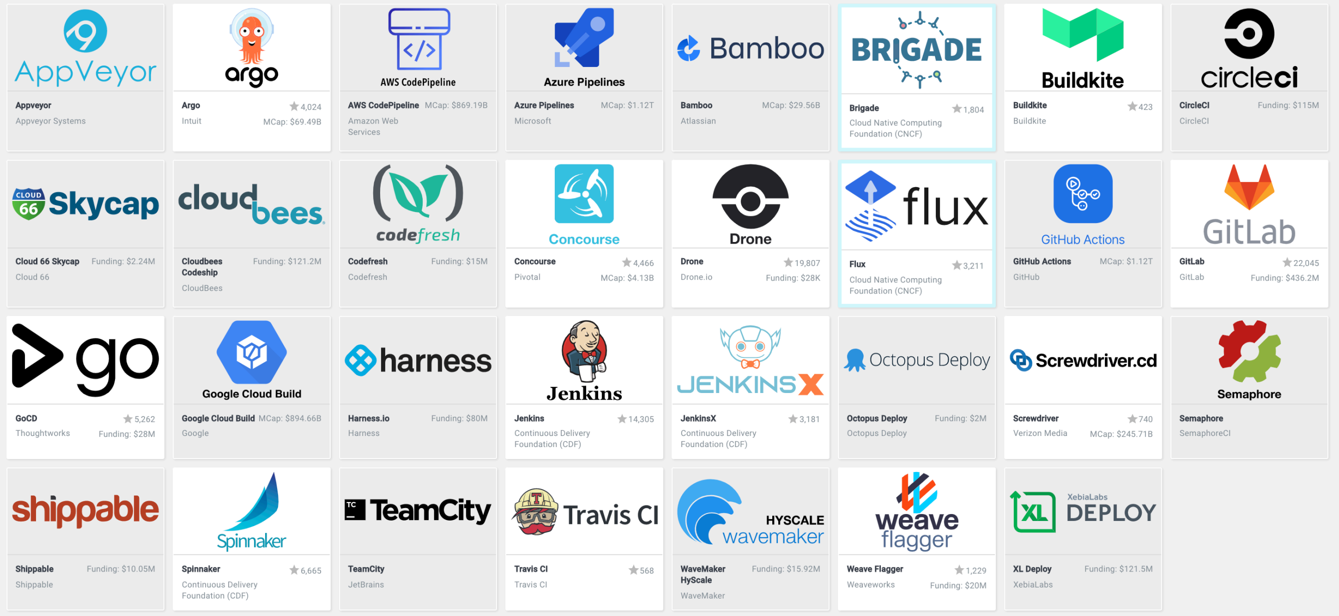
Tekton Concepts and Examples

Tekton Features

Tekton and Kabanero

Tekton provides Kubernetes-style resources for declaring CI/CD concepts

Industry Challenges



Appveyor
Appveyor Systems

Argo ★4,024
Intuit MCap: $69.49B

AWS CodePipeline MCap: $869.19B
Amazon Web Services

Azure Pipelines MCap: $1.12T
Microsoft

Bamboo MCap: $29.56B
Atlassian

Brigade ★1,804
Cloud Native Computing Foundation (CNCF)

Buildkite ★423
Buildkite

CircleCI Funding: $115M
CircleCI

Cloud 66 Skycap Funding: $2.24M
Cloud 66

Cloudbees Codeship Funding: $121.2M
CloudBees

Codefresh Funding: $15M
Codefresh

Concourse ★4,466
Pivotal MCap: $4.13B

Drone ★19,807
Drone.io Funding: $28K

Flux ★3,211
Cloud Native Computing Foundation (CNCF)

GitHub Actions MCap: $1.12T
GitHub

GitLab ★22,045
GitLab Funding: $436.2M

GoCD ★5,262
Thoughtworks Funding: $28M

Google Cloud Build MCap: $894.66B
Google

Harness.io Funding: $80M
Harness

Jenkins ★14,305
Continuous Delivery Foundation (CDF)

JenkinsX ★3,181
Continuous Delivery Foundation (CDF)

Octopus Deploy Funding: $2M
Octopus Deploy

Screwdriver ★740
Verizon Media MCap: $245.71B

Semaphore
SemaphoreCI

Shippable Funding: $10.05M
Shippable

Spinnaker ★6,665
Continuous Delivery Foundation (CDF)

TeamCity
JetBrains

Travis CI ★568
Travis CI

WaveMaker HyScale Funding: $15.92M
WaveMaker

Weave Flagger ★1,229
Weaveworks Funding: $20M

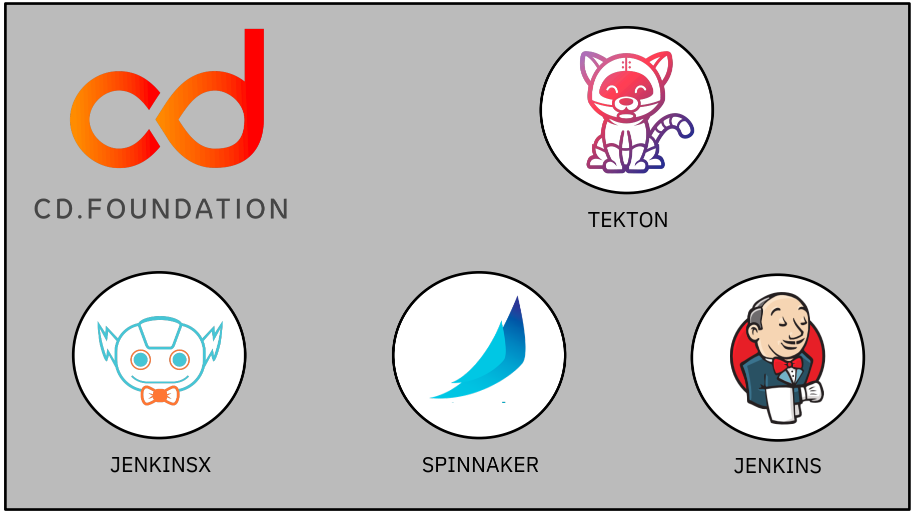
XL Deploy Funding: $121.5M
XebiaLabs

- Competing projects
- Conflicting Terminology
- Achieving the same goal

Image from CNCF Landscape: https://landscape.cncf.io/

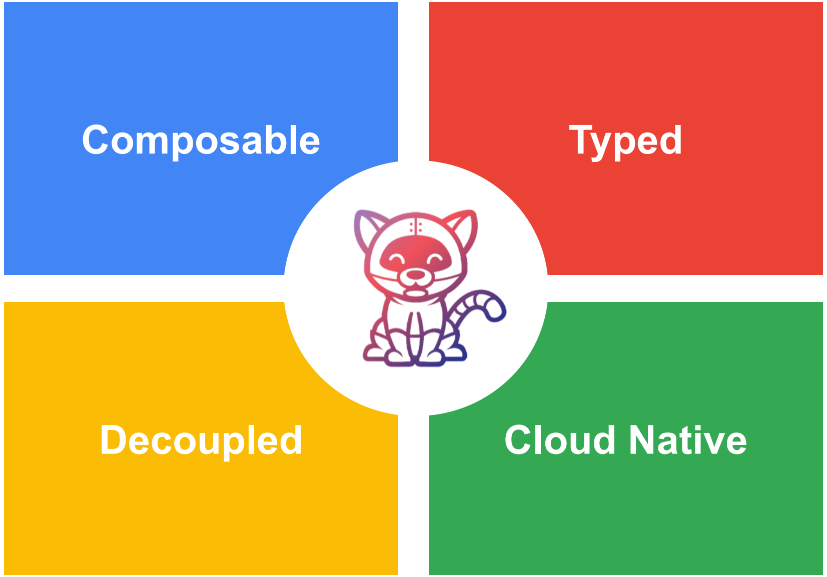
Tekton Emerges

- Spun out of the Knative build project
- Git Repo: https://github.com/tektoncd
- **Current release**: 0.10.0 (January 2020)
- **First release**: 0.2.0 (March 2019)
- **Contributors**: Google, Red Hat, Pivotal, IBM, etc.
- Part of the **CD Foundation** (under the Linux Foundation)
  - Includes other open source projects such as: JenkinsX, Jenkins, Spinnaker, Tekton
- CD Foundation announced in March 2019
  - **Goal**: To serve as the vendor-neutral home for the most important open source projects for continuous delivery



So, what is Tekton?

- **Cloud Native**: Run on Kubernetes, has Kubernetes clusters as a first-class type, use containers as their building blocks
- **Typed**: The concept of typed resources means that for a resources can swap out implementations
- **Composable**: Tekton concepts build upon each other
- **Decoupled**: The Tasks which make up a Pipeline can easily be run in isolation. One Pipeline can be used to deploy to any k8s cluster



Composable

Typed

Decoupled

Cloud Native

Tekton Project Goals

Goals

- Specify the “API” and provide the “building blocks” for running build pipelines
- Host a community of sub-projects that extend Tekton (Dashboard UI, CLI, Webhooks, etc.)
- Provide a catalog of best practices for authoring pipelines and tasks

Tekton Concept: Step

- The smallest building block
- Specify images, commands, arguments
- Is a container

```yaml
steps:
  - name: echo
    image: ubuntu
    command:
      - echo
    args:
      - "hello world"
```

Tekton CRD: Task

- New CRD
- Sequence of **Steps**
- Run in sequential order
- Reusable
- Perform a specific task
- Runs on the same k8s node

```yaml
apiVersion: tekton.dev/v1alpha1
kind: Task
metadata:
  name: echo-hello-world
spec:
  steps:
    - name: echo
      image: ubuntu
      command:
        - echo
      args:
        - "hello world"
```

Tekton CRD: Pipeline

- Expresses **Tasks**
  - Sequentially
  - Concurrently
- Links input and output
- Execute **Tasks** on different nodes

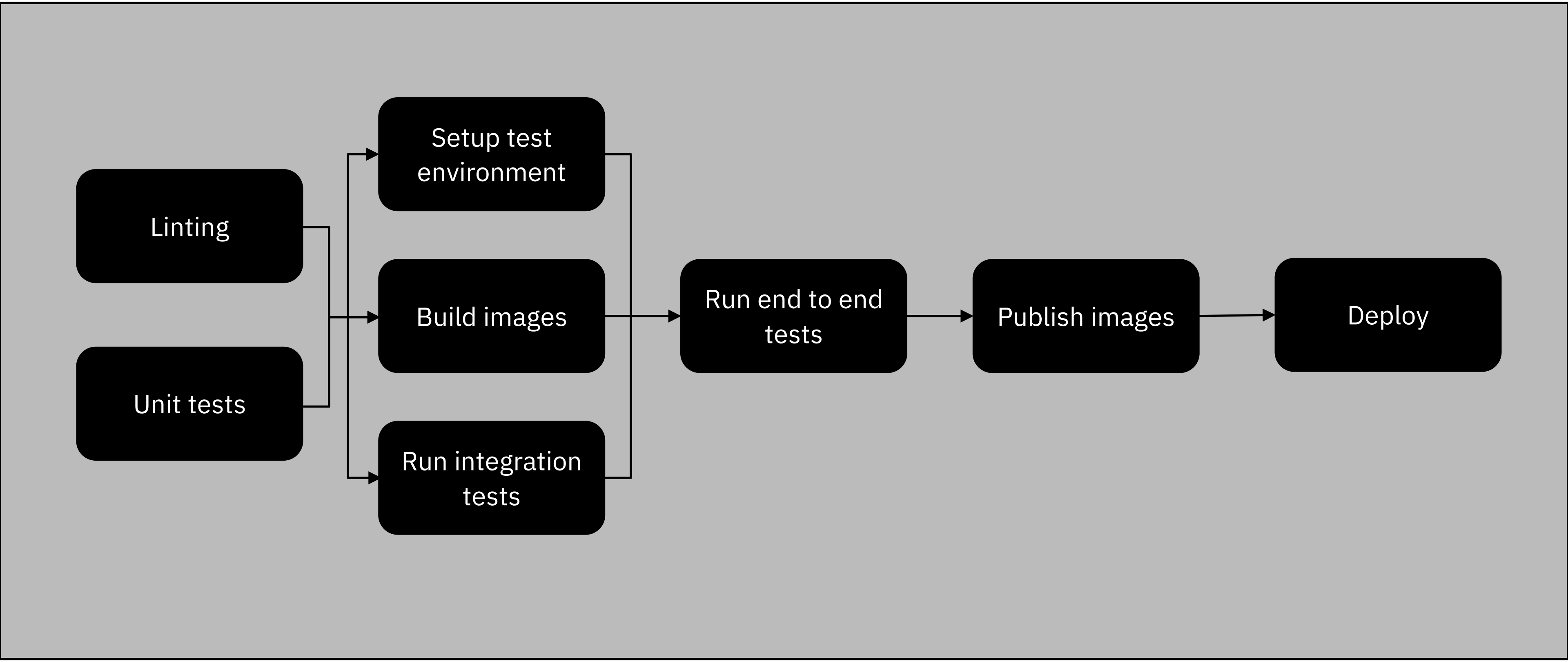
```
apiVersion: tekton.dev/v1alpha1
kind: Pipeline
metadata:
 name: tutorial-pipeline
spec:
 - name: build-app
  taskRef:
   name: build-push
  resources:
   outputs:
    - name: image
     resource: my-image
 - name: deploy-app
  taskRef:
   name: deploy-kubectl
  resources:
   inputs:
    - name: image
   resource: my-image
   from:
    - build-app
```

Tekton Runtime CRDs

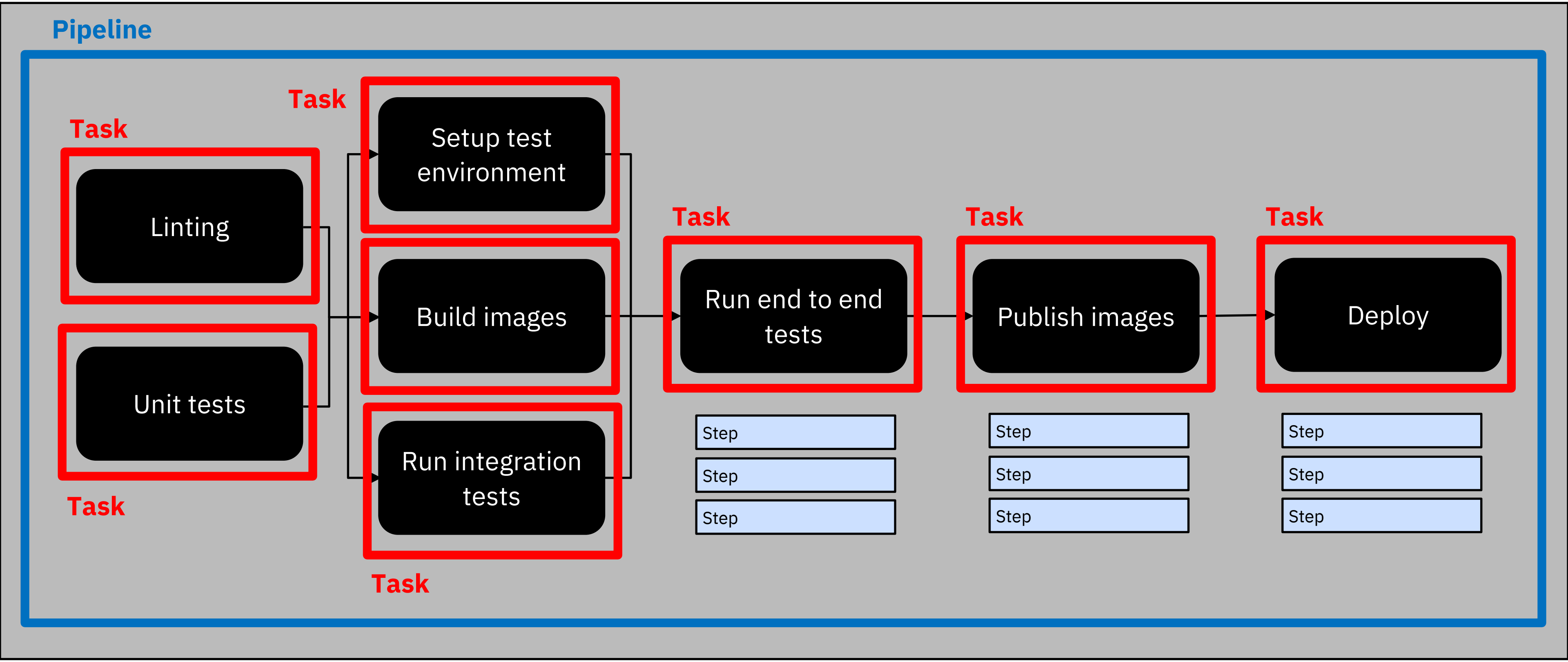
- Instances of Pipelines and Tasks:
  - PipelineRun
  - TaskRun
- Runtime info such as registry information and git repo

```yaml
apiVersion: tekton.dev/v1alpha1
kind: PipelineRun
metadata:
  name: tutorial-pipeline-run-1
spec:
  serviceAccountName: tutorial-service
  pipelineRef:
    name: tutorial-pipeline
  resources:
    - name: source-repo
      resourceRef:
        name: skaffold-git
    - name: web-image
      resourceRef:
        name: skaffold-image-leeroy-web
```

Putting it all together

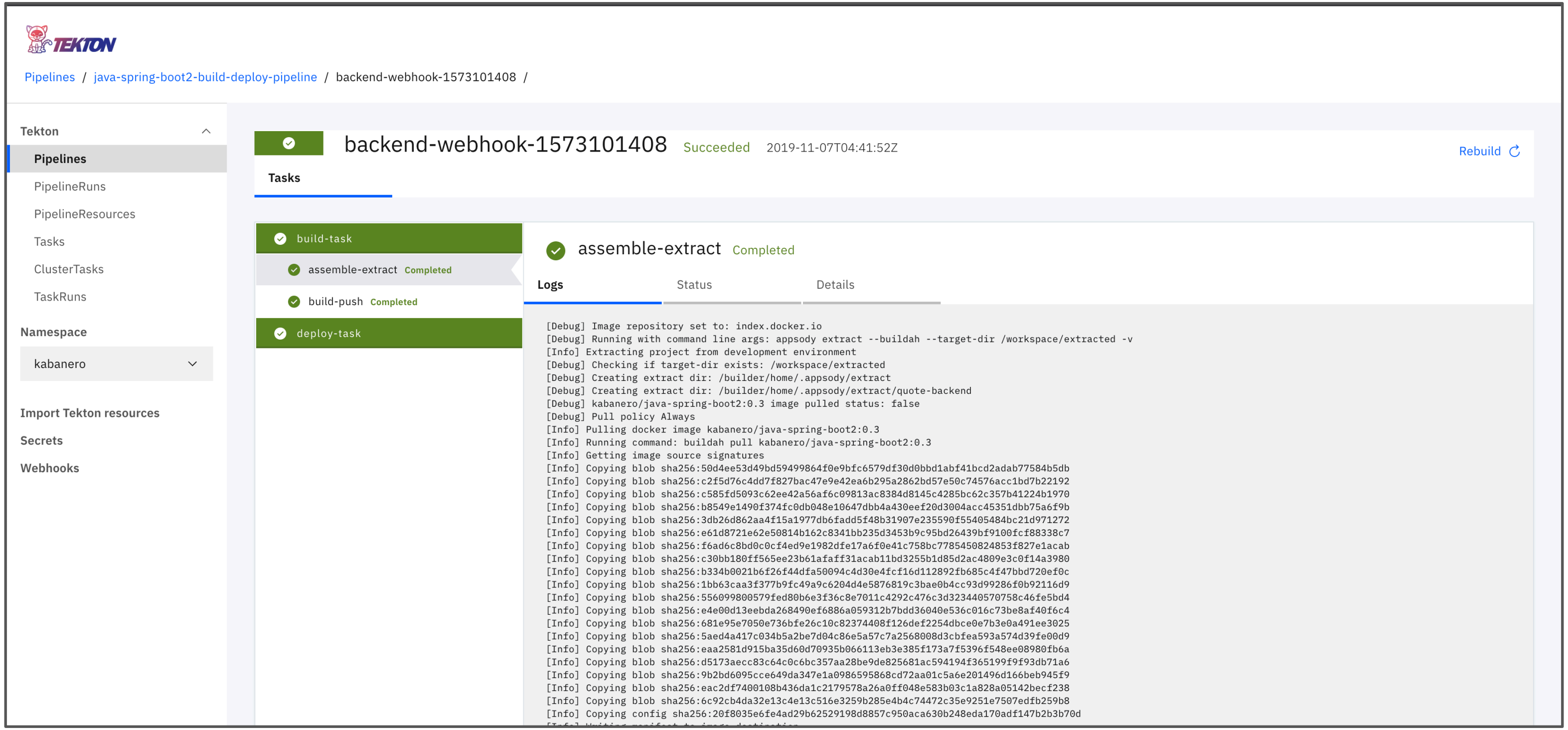


Putting it all together



Tekton Features: Dashboard

https://github.com/tektoncd/dashboard

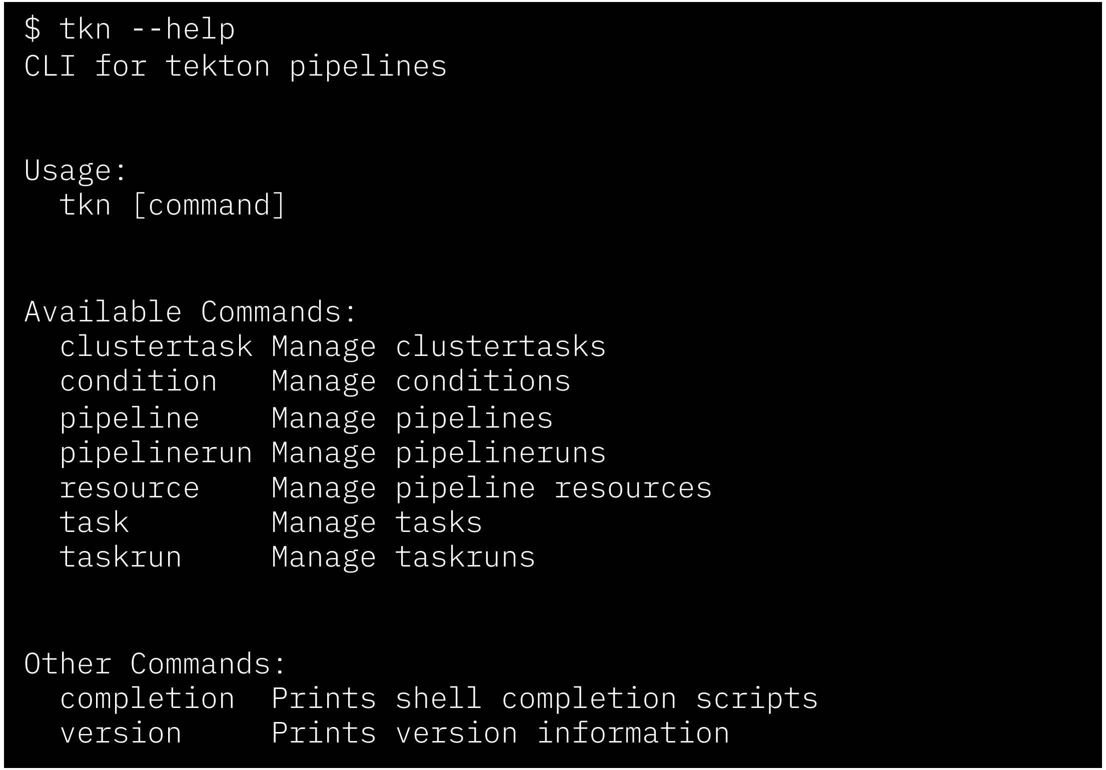


```
[Debug] Image repository set to: index.docker.io
[Debug] Running with command line args: appsody extract --buildah --target-dir /workspace/extracted -v
[Info] Extracting project from development environment
[Debug] Checking if target-dir exists: /workspace/extracted
[Debug] Creating extract dir: /builder/home/.appsody/extract
[Debug] Creating extract dir: /builder/home/.appsody/extract/quote-backend
[Debug] kabanero/java-spring-boot2:0.3 image pulled status: false
[Debug] Pull policy Always
[Info] Pulling docker image kabanero/java-spring-boot2:0.3
[Info] Running command: buildah pull kabanero/java-spring-boot2:0.3
[Info] Getting image source signatures
[Info] Copying blob sha256:50d4ee53d49bd59499864f0e9bfc6579df30d0bbd1abf41bcd2adab77584b5db
[Info] Copying blob sha256:c2f5d76c4dd7f827bac47e9e42ea6b295a2862bd57e50c74576acc1bd7b22192
[Info] Copying blob sha256:c585fd5093c62ee42a56af6c09813ac8384d8145c4285bc62c357b41224b1970
[Info] Copying blob sha256:b8549e1490f374fc0db048e10647dbb4a430eef20d3004acc45351dbb75a6f9b
[Info] Copying blob sha256:3db26d862aa4f15a1977db6fadd5f48b31907e235590f55405484bc21d971272
[Info] Copying blob sha256:e61d8721e62e50814b162c8341bb235d3453b9c95bd26439bf9100fcf88338c7
[Info] Copying blob sha256:f6ad6c8bd0c0cf4ed9e1982dfe17a6f0e41c758bc7785450824853f827e1acab
[Info] Copying blob sha256:c30bb180ff565ee23b61afaff31acab11bd3255b1d85d2ac4809e3c0f14a3980
[Info] Copying blob sha256:b334b0021b6f26f44dfa50094c4d30e4fcf16d112892fb685c4f47bbd720ef0c
[Info] Copying blob sha256:1bb63caa3f377b9fc49a9c6204d4e5876819c3bae0b4cc93d99286f0b92116d9
[Info] Copying blob sha256:556099800579fed80b6e3f36c8e7011c4292c476c3d323440570758c46fe5bd4
[Info] Copying blob sha256:e4e00d13eebda268490ef6886a059312b7bdd36040e536c016c73be8af40f6c4
[Info] Copying blob sha256:681e95e7050e736bfe26c10c82374408f126def2254dbce0e7b3e0a491ee3025
[Info] Copying blob sha256:5aed4a417c034b5a2be7d04c86e5a57c7a2568008d3cbfea593a574d39fe00d9
[Info] Copying blob sha256:eaa2581d915ba35d60d70935b066113eb3e385f173a7f5396f548ee08980fb6a
[Info] Copying blob sha256:d5173aecc83c64c0c6bc357aa28be9de825681ac594194f365199f9f93db71a6
[Info] Copying blob sha256:9b2bd6095cce649da347e1a0986595868cd72aa01c5a6e201496d166beb945f9
[Info] Copying blob sha256:eac2df7400108b436da1c2179578a26a0ff048e583b03c1a828a05142becf238
[Info] Copying blob sha256:6c92cb4da32e13c4e13c516e3259b285e4b4c74472c35e9251e7507edfb259b8
[Info] Copying config sha256:20f8035e6fe4ad29b62529198d8857c950aca630b248eda170adf147b2b3b70d
[Info] Writing manifest to image destination
```

Output from a PipelineRun

Tekton Features: CLI

https://github.com/tektoncd/cli

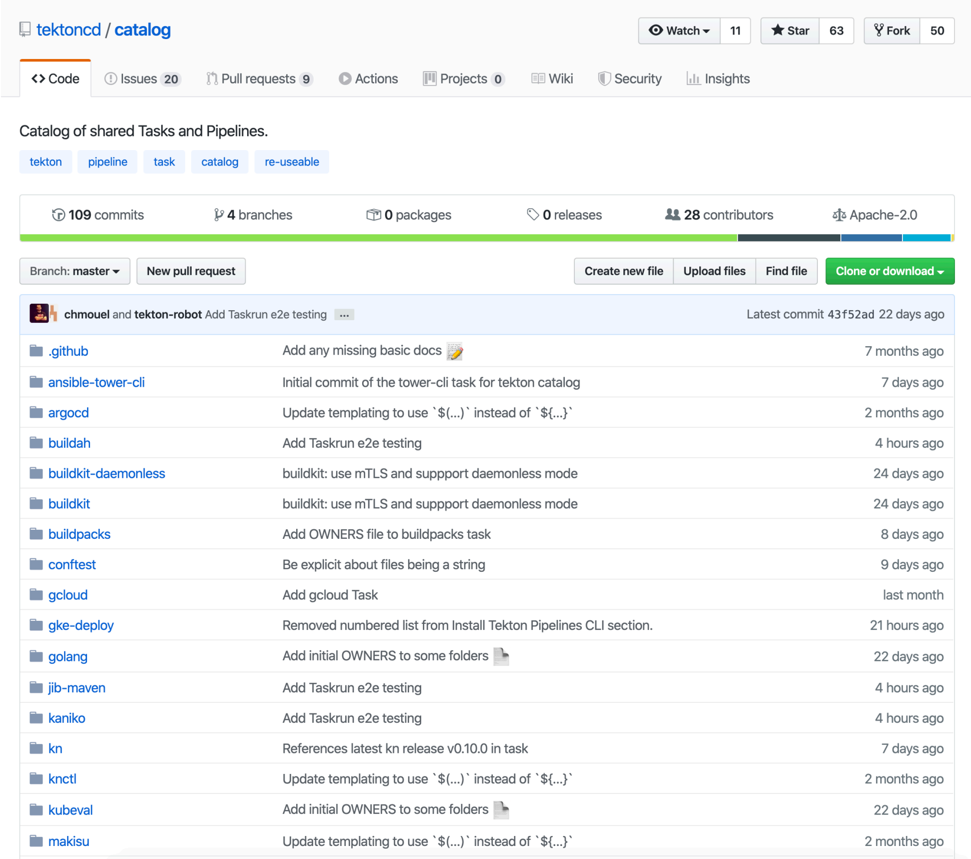


```
$ tkn --help
CLI for tekton pipelines


Usage:
  tkn [command]


Available Commands:
  clustertask Manage clustertasks
  condition   Manage conditions
  pipeline    Manage pipelines
  pipelinerun Manage pipelineruns
  resource    Manage pipeline resources
  task        Manage tasks
  taskrun     Manage taskruns


Other Commands:
  completion  Prints shell completion scripts
  version     Prints version information
```

Tekton Features: Catalog

https://github.com/tektoncd/catalog

This repository contains a catalog of **Task** resources, designed to be reusable in many pipelines.



Catalog of shared Tasks and Pipelines.

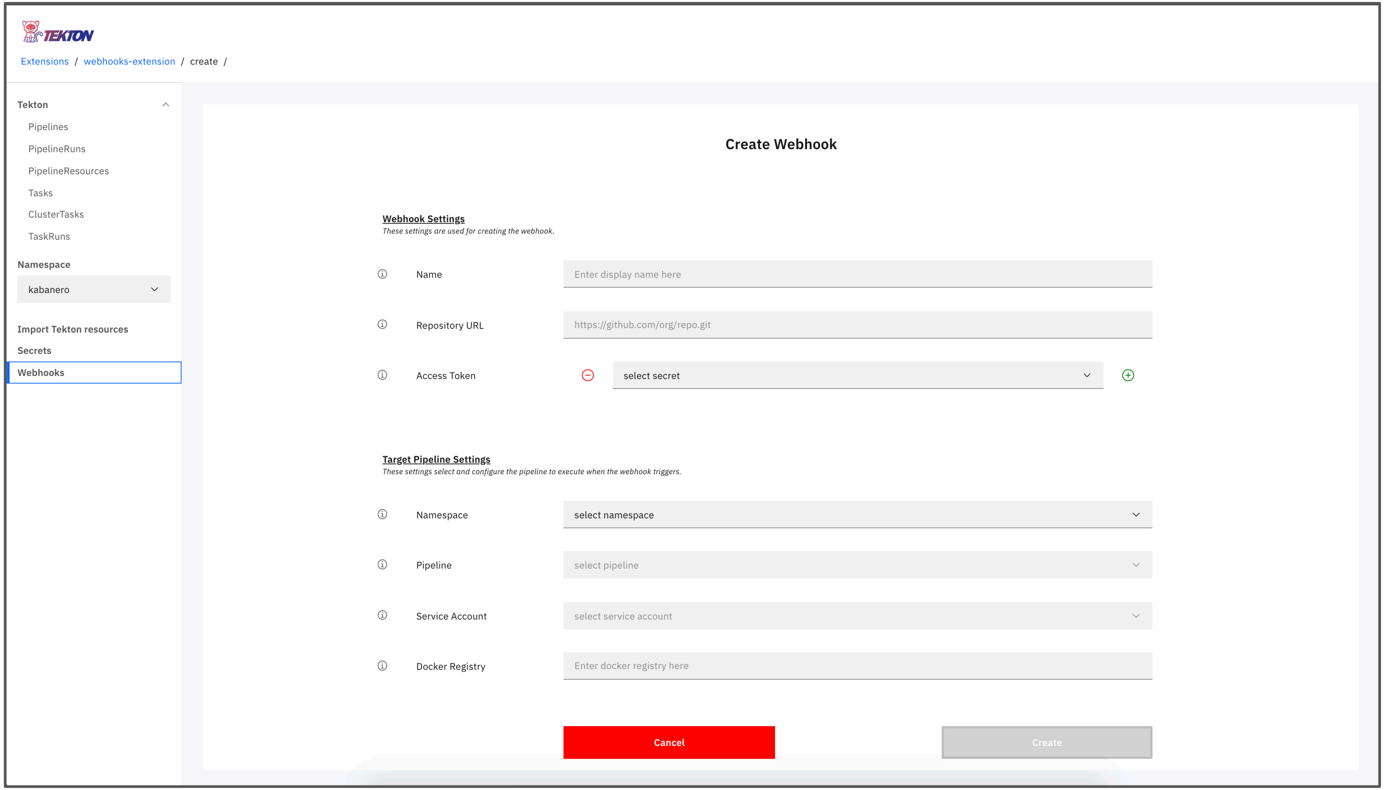
chmouel and tekton-robot Add Taskrun e2e testing ... Latest commit 43f52ad 22 days ago

.github	Add any missing basic docs	7 months ago
ansible-tower-cli	Initial commit of the tower-cli task for tekton catalog	7 days ago
argocd	Update templating to use `$(...)` instead of `${...}`	2 months ago
buildah	Add Taskrun e2e testing	4 hours ago
buildkit-daemonless	buildkit: use mTLS and suppport daemonless mode	24 days ago
buildkit	buildkit: use mTLS and suppport daemonless mode	24 days ago
buildpacks	Add OWNERS file to buildpacks task	8 days ago
conftest	Be explicit about files being a string	9 days ago
gcloud	Add gcloud Task	last month
gke-deploy	Removed numbered list from Install Tekton Pipelines CLI section.	21 hours ago
golang	Add initial OWNERS to some folders	22 days ago
jib-maven	Add Taskrun e2e testing	4 hours ago
kaniko	Add Taskrun e2e testing	4 hours ago
kn	References latest kn release v0.10.0 in task	7 days ago
knctl	Update templating to use `$(...)` instead of `${...}`	2 months ago
kubeval	Add initial OWNERS to some folders	22 days ago
makisu	Update templating to use `$(...)` instead of `${...}`	2 months ago

Tekton Features: Webhooks

https://github.com/tektoncd/experimental/tree/master/webhooks-extension

Allows users to set up GitHub webhooks that will trigger **PipelineRuns** and **TaskRuns**.



Create Webhook

Webhook Settings

These settings are used for creating the webhook.

Name: Enter display name here

Repository URL: https://github.com/org/repo.git

Access Token: select secret

Target Pipeline Settings

These settings select and configure the pipeline to execute when the webhook triggers.

Namespace: select namespace

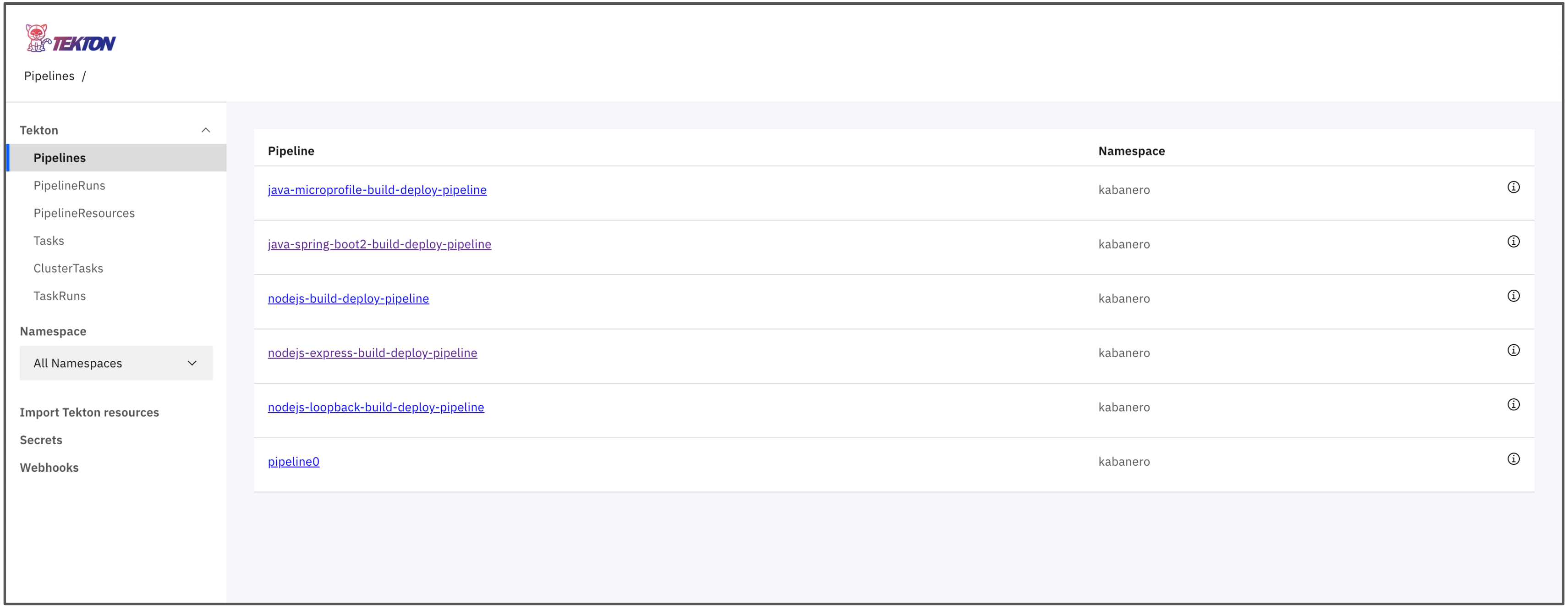
Pipeline: select pipeline

Service Account: select service account

Docker Registry: Enter docker registry here

Tekton with Kabanero

https://github.com/kabanero-io/kabanero-pipelines



Pipeline	Namespace
java-microprofile-build-deploy-pipeline	kabanero
java-spring-boot2-build-deploy-pipeline	kabanero
nodejs-build-deploy-pipeline	kabanero
nodejs-express-build-deploy-pipeline	kabanero
nodejs-loopback-build-deploy-pipeline	kabanero
pipeline0	kabanero

- Each Stack in Kabanero comes with a default pipeline (named <lang>-build-deploy).
- The pipeline consists of two tasks: One to build the appsody stack, the second to deploy the appsody stack