

Tekton Overview

Client Developer Advocacy

Contents

History of Tekton

What is Tekton?

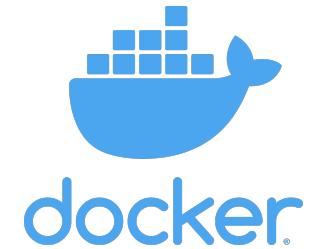
Tekton Concepts and Examples

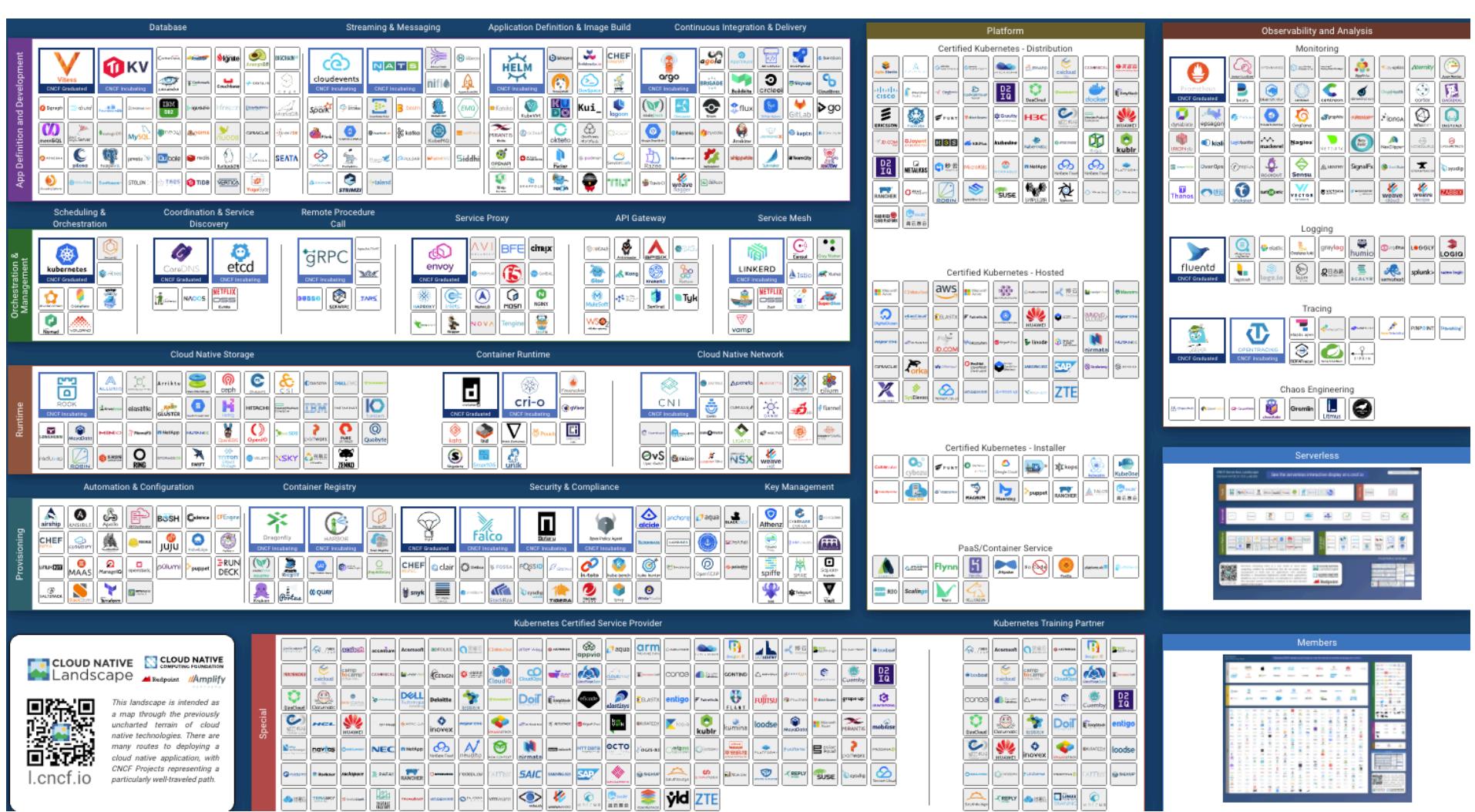
Tekton Features

Tekton and Kabanero



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





Apache License 2.0 (12)

 Agola ★ 536 Sorint.Lab	 Argo ★ 5,255 Cloud Native Computing Foundation (CNCF)	 Brigade ★ 1,924 Cloud Native Computing Foundation (CNCF)	 Concourse ★ 4,871 VMware MCap: \$53.51B	 Flux ★ 4,357 Cloud Native Computing Foundation (CNCF)	 GoCD ★ 5,560 Thoughtworks Funding: \$28M
 JenkinsX ★ 3,416 Continuous Delivery Foundation (CDF)	 Keptn ★ 370 Dynatrace MCap: \$7.66B	 Razee ★ 325 IBM MCap: \$106B	 Spinnaker ★ 7,021 Continuous Delivery Foundation (CDF)	 Tekton Pipelines ★ 4,771 Continuous Delivery Foundation (CDF)	 Weave Flagger ★ 1,808 Weaveworks Funding: \$20M

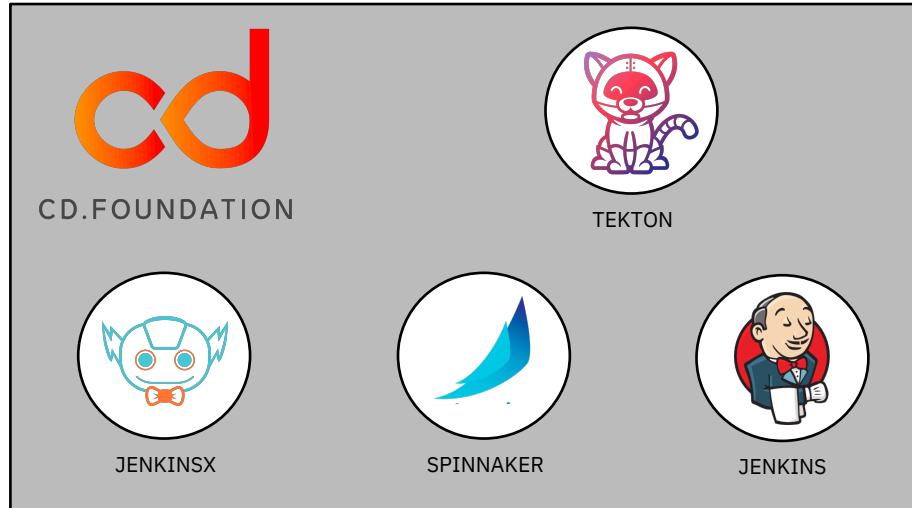
MIT License (3)

 Buildkite ★ 460 Buildkite	 Jenkins ★ 15,321 Continuous Delivery Foundation (CDF)	 Travis CI ★ 582 Travis CI
--	--	--

History of Tekton

<http://ibm.biz/tekton-lab>

- Originates from Knative build Project
- Git Repo: <https://github.com/tektoncd>
- Docs: <https://tekton.dev/docs/>
- First release: 0.1.0 (February 20, 2019)
- Current release: 0.11.3 (April 24, 2020)
- Contributors: IBM, Red Hat, Google, Cloudbees, Pivotal, etc.
- Integrated with Knative, Triggermesh Aktion, Jenkins X
- 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

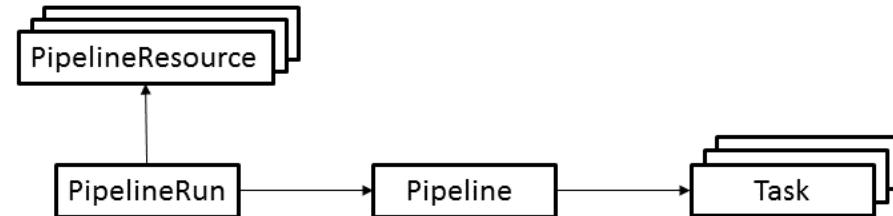


Tekton

<http://ibm.biz/tekton-lab>

Tekton is implemented as a set of Kubernetes Custom Resource Definitions (CRDs).

- A **PipelineResource** defines an object that is an input (such as a git repository) or an output (such as a docker image) of the pipeline.
- A **PipelineRun** defines an execution of a pipeline. It references the **Pipeline** to run and the **PipelineResources** to use as inputs and outputs.
- A **Pipeline** defines the set of **Tasks** that compose a pipeline.
- A **Task** defines a set of build steps such as compiling code, running tests, and building and deploying images.



Tekton supports many different types of resources, including:
git:

- git repository
- Pull Request: specific pull request in a git repository
- Image: container image
- Cluster: Kubernetes cluster
- Storage: object or directory in a blob store,
- CloudEvent: CloudEvent

Tekton

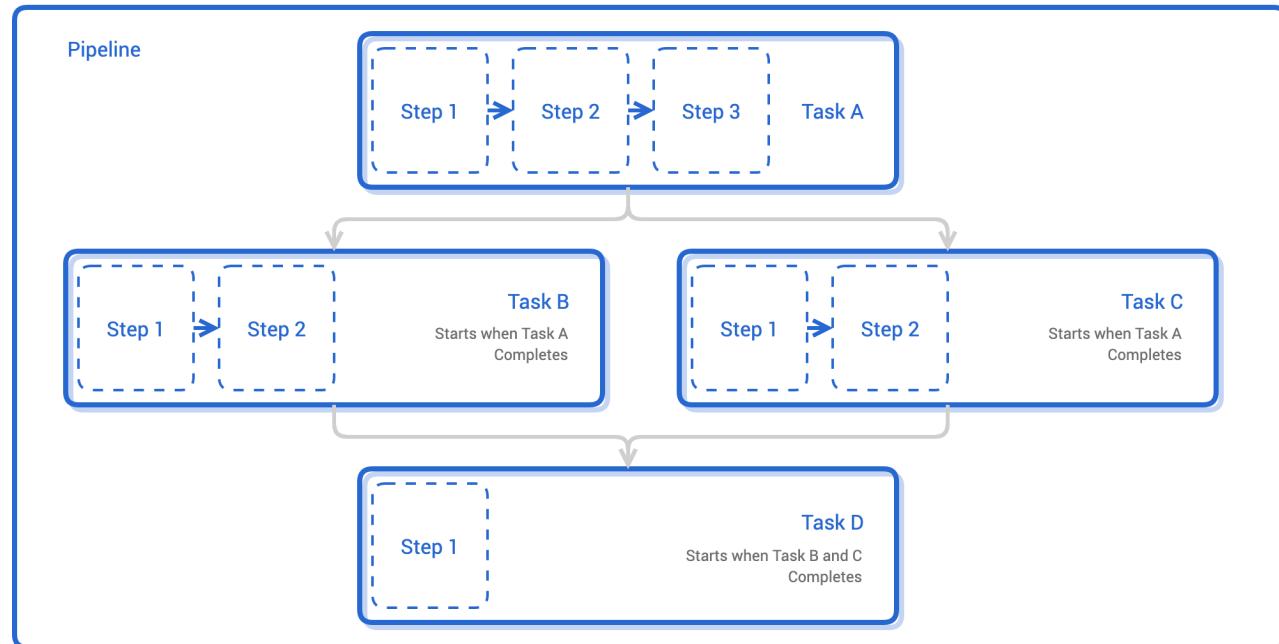
<http://ibm.biz/tekton-lab>

Tekton is optimized for building and deploying cloud-native.

Tekton runs a task as a Kubernetes pod. Each step is a running container in the pod.

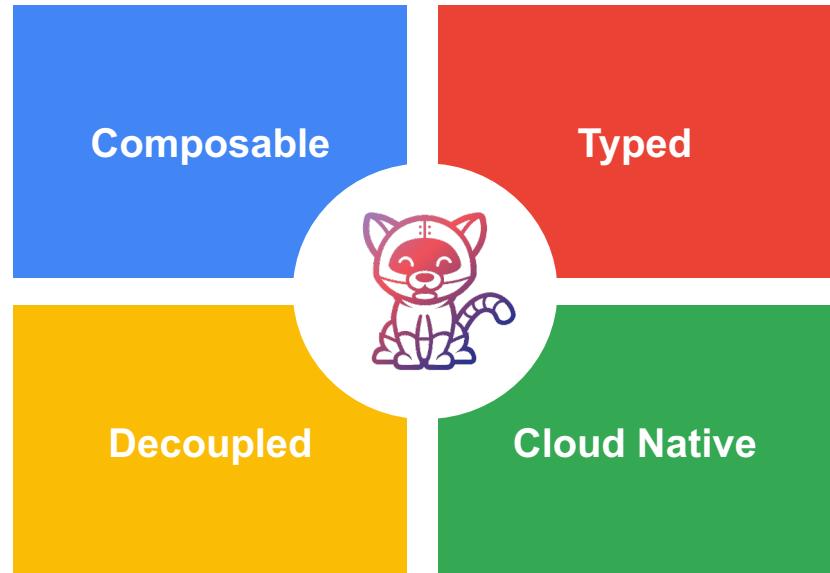
This allows you to create a shared environment for related steps.

Tekton Pipelines schedule some built-in containers to run automatically before and after step containers.



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 resource 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



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

```
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

```
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
```

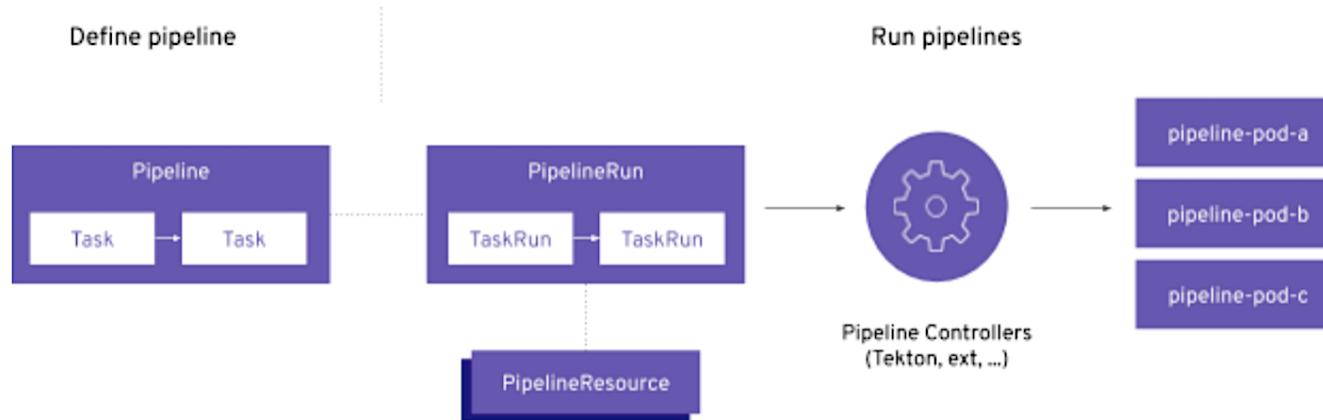
Tekton Runtime CRDs

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

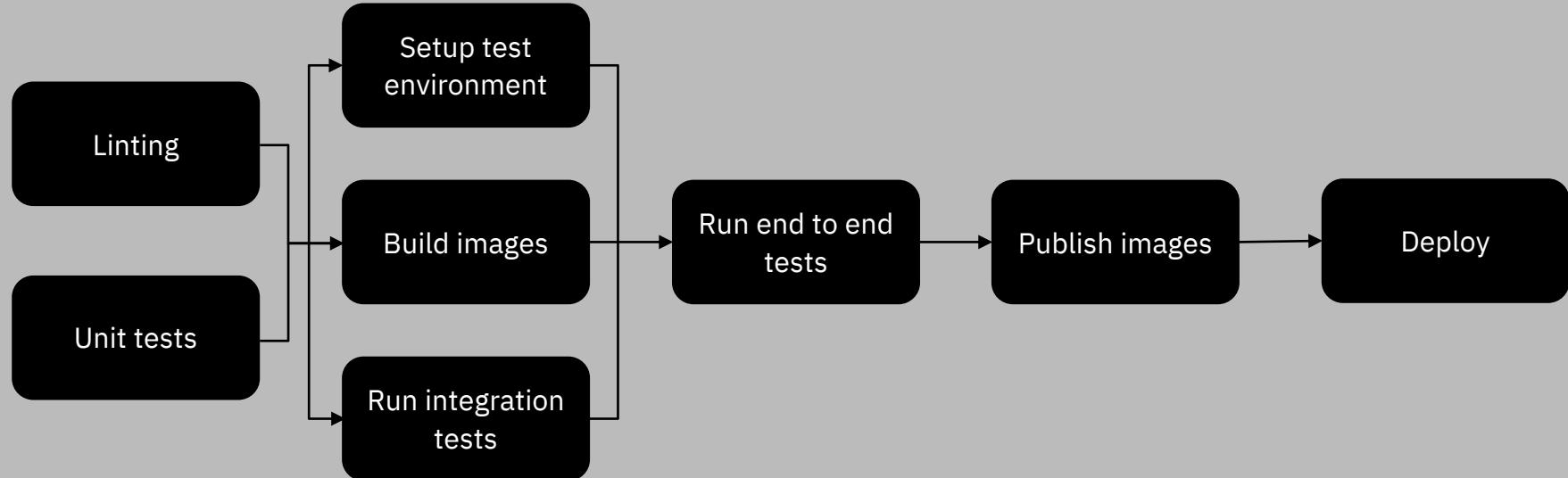
```
apiVersion: tekton.dev/v1alpha1
kind: PipelineRun
metadata:
  name: tutorial-pipeline-run-1
spec:
  serviceAccountName: tutorial-service
  pipelineRef:
    name: tutorial-pipeline
  resources:
    - name: source-repo
      resourceRef:
        name: skaffold-git
    - name: web-image
      resourceRef: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

Tekton

<http://ibm.biz/tekton-lab>

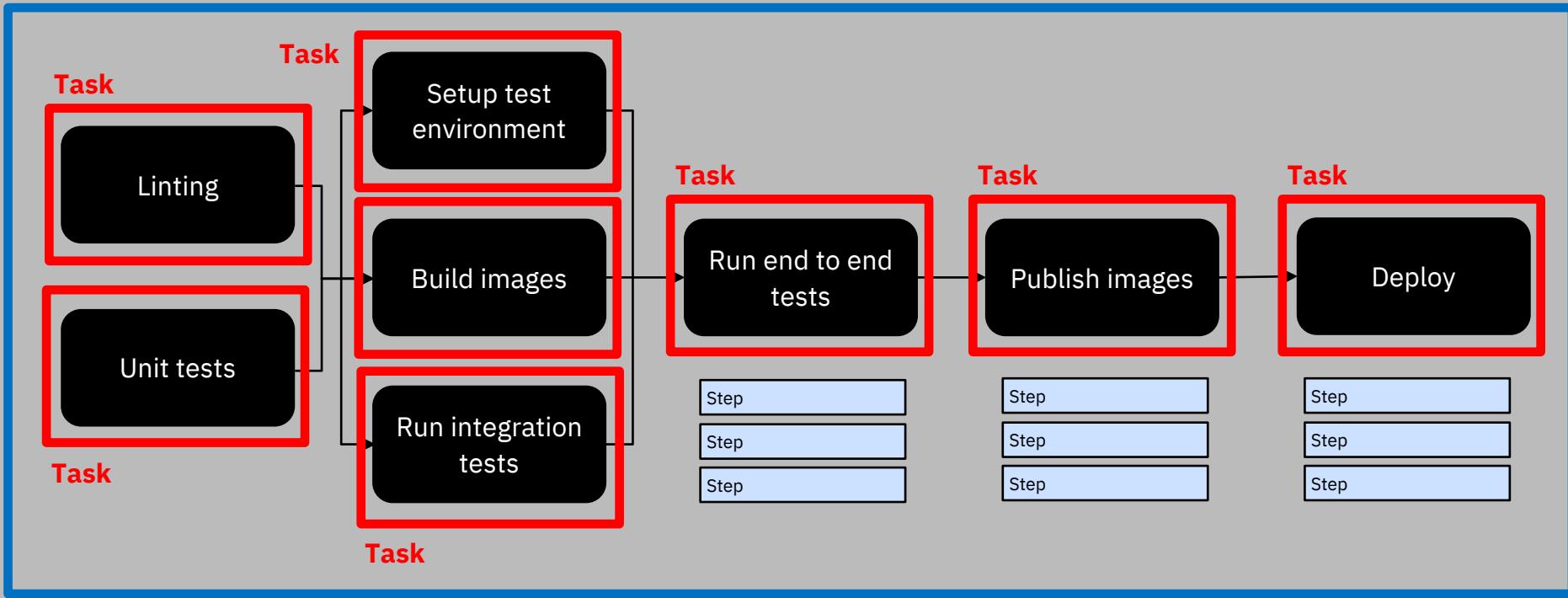


Putting it all together



Putting it all together

Pipeline



Tekton Features: Dashboard

<https://github.com/tektoncd/dashboard>

The screenshot shows the Tekton Dashboard interface. On the left, there's a sidebar with navigation links: Pipelines, PipelineRuns, PipelineResources, Tasks, ClusterTasks, TaskRuns, Namespace (selected), and Import Tekton resources, Secrets, Webhooks. The main area displays a PipelineRun titled "backend-webhook-1573101408" which has "Succeeded" status and was run on 2019-11-07T04:41:52Z. It lists four tasks: "build-task" (Completed), "assemble-extract" (Completed), "build-push" (Completed), and "deploy-task" (Completed). The "assemble-extract" task is expanded, showing its logs. The logs output the command "aspyd extract --buildah --target-dir /workspace/extracted -v" and various debug messages related to extracting the project from the development environment, creating extract dirs, and pulling the image.

```
[Debug] Image repository set to: index.docker.io
[Info] Running with command line args: aspyd 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/.aspyd/extract
[Debug] Creating extract dir: /builder/home/.aspyd/extract/quote-backend
[Debug] kabanero/java-spring-boot2:0.3 image pulled status: false
[Debug] Pull policy Always
[Debug] Pull policy Always
[Info] Pulling docker image kabanero/java-spring-boot:0.3
[Info] Running command: buildah pull kabanero/java-spring-boot:0.3
[Info] Getting image source signature
[Info] Copying blob sha256:504de453d49bd59499864f0e9bf6579df30d0bb1ab41bcd2adab77584b5db
[Info] Copying blob sha256:c25d76c4d7f827bac47fe942ea6b295a2862bd57e56c74576acc4537b1bd7b22192
[Info] Copying blob sha256:c5851d593122e242a5e6fc09813ac8384d81a5c4285bdc2357b41224b1970
[Info] Copying blob sha256:1884974901374fc0b048610647d04a390ee12083994ac45357b5a659b
[Info] Copying blob sha256:1884974901374fc0b048610647d04a390ee12083994ac45357b5a659b
[Info] Copying blob sha256:1884974901374fc0b048610647d04a390ee12083994ac45357b5a659b
[Info] Copying blob sha256:e14d0721a2e509a1b16c83414b25a5453b5e93b2a2439bf19400e198311272
[Info] Copying blob sha256:1b63ca3c3f7709fc499e62044e58768193bae0b4cc93ff9286f69b211649
[Info] Copying blob sha256:55a0998808579fe180b6e313e8e761124922476c3d323440570756c46fe5bd4
[Info] Copying blob sha256:e4e00d13eebda2684909ef886a059312b7bd3d6040e0536e016c73be8af40f16c4
[Info] Copying blob sha256:681e957850e736bf62e6c10c8237408f126de12254dbc0e703e0a491ee3025
[Info] Copying blob sha256:5aa4d4a17c034b5a2be7d4c86e5a5a57c7a25680883d3cbfe0a593a574439fe09d9
[Info] Copying blob sha256:ea2581a915ba356d0679935b6e6113eb3e385f173a7f5396f548e08980fb6a
[Info] Copying blob sha256:d573aeca833c64c0c6bc357a28be9d825681a5c94194f365199f9f93db71a6
[Info] Copying blob sha256:92bd6095cce649da347e7a0986595868c7d2a015c6e201496d166be0945f9
[Info] Copying blob sha256:ea2f74001080a43da1c2179578a2668c048e583b03c1a828a05142b4f238
[Info] Copying blob sha256:6c92c04432e13c4e13c516e3259b285e04c74472c35e9251a7507edfb2598
[Info] Copying config sha256:20f8035e6fe4ad2962529198d8857c950aca630b248eda170adf147b2b3b070d
```

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>

The screenshot shows the GitHub repository page for 'tektoncd/catalog'. At the top, it displays basic repository statistics: 109 commits, 4 branches, 0 packages, 0 releases, 28 contributors, and Apache-2.0 license. Below this, a list of recent commits is shown, each with a small icon, the author's name, a brief description, and the time since the commit.

Author	Commit Message	Time Ago
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 support daemonless mode	24 days ago
buildkit	buildkit: use mTLS and support 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

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

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**.

The screenshot shows the 'Create Webhook' form in the Tekton UI. On the left, a sidebar lists 'Tekton' resources: Pipelines, PipelineRuns, PipelineResources, Tasks, ClusterTasks, TaskRuns, Namespace (set to 'kabanero'), Import Tekton resources, Secrets, and Webhooks (which is selected and highlighted with a blue border). The main area is titled 'Create Webhook'. It contains two sections: 'Webhook Settings' and 'Target Pipeline Settings'. In 'Webhook Settings', there are fields for 'Name' (placeholder: 'Enter display name here'), 'Repository URL' (placeholder: 'https://github.com/org/repo.git'), and 'Access Token' (placeholder: 'select secret'). A '+' icon is available to add more secrets. In 'Target Pipeline Settings', there are fields for 'Namespace' (placeholder: 'select namespace'), 'Pipeline' (placeholder: 'select pipeline'), 'Service Account' (placeholder: 'select service account'), and 'Docker Registry' (placeholder: 'Enter docker registry here'). At the bottom right are 'Cancel' and 'Create' buttons.

