Cloud Native CI/CD

with Jenkins X and ~~Knative~~ Tekton Pipelines

@jdrawlings

@bobcatwilson


@jdrawlings
@bobcatwilson

Review requested
Review has been requested on this pull request. Learn more.

Show all reviewers

Some checks were not successful
1 failing, 1 pending, and 4 successful checks

Hide all checks

- ✕ **integration-tests** — Job failed. Required Details
- ✓ **coverage** — Job succeeded. Details
- ✓ **unit-tests** — Job succeeded. Required Details

Required statuses must pass before merging
All required statuses and check runs on this pull request must run successfully to enable automatic merging.

Rebase and merge ▾ You can also open this in GitHub Desktop or view command line instructions.

@jdrawlings

@bobcatwilson

```
➜  test git:(master) ✗ ./integration-tests.sh
+ container-diff diff us.gcr.io/catfactory-production/image
daemon://us.gcr.io/catfactory-production/image
./integration-tests.sh: line 472: container-diff: command not found
```

@jdrawlings

@bobcatwilson

```
➜  test git:(master) ✗ ./integration-tests.sh
+ docker push us.gcr.io/catfactory-production/image
The push refers to repository [us.gcr.io/catfactory-production/image]
d250020dd5b7: Waiting
503e53e365f3: Waiting
denied: requested access to the resource is denied
```

@jdrawlings

@bobcatwilson

```
-docker build -t us.gcr.io/catfactory-production/image -f images/Dockerfile images/
-docker push us.gcr.io/catfactory-production/image
+docker build -t us.gcr.io/christies-image-registry/image -f images/Dockerfile images/
+docker push us.gcr.io/christies-image-registry/image
```

@jdrawlings

@bobcatwilson

```
➜  test git:(master) ✗ ./integration-tests.sh
+ kubectl apply -f newly-built.yaml
Error from server (Forbidden): error when creating "newly-build.yaml":...
```

@jdrawlings

@bobcatwilson


@jdrawlings
@bobcatwilson

We can do better!

@jdrawlings

@bobcatwilson

James Rawlings

Software Engineer at Cloudbees
Co-creator of Jenkins X

Christie Wilson

Software Engineer at Google
Tekton Pipelines Lead

Jenkins X + Tekton Pipelines = Cloud Native CI/CD

@jdrawlings

@bobcatwilson

Cloud Native CI/CD > bash CI/CD

@jdrawlings

@bobcatwilson

What’s this “cloud native”?

@jdrawlings

@bobcatwilson

Cloud Native

(As defined by the CNCF)

“Cloud native computing uses an open source software stack to deploy applications as microservices, packaging each part into its own container, and dynamically orchestrating those containers to optimize resource utilization.”

Cloud Native

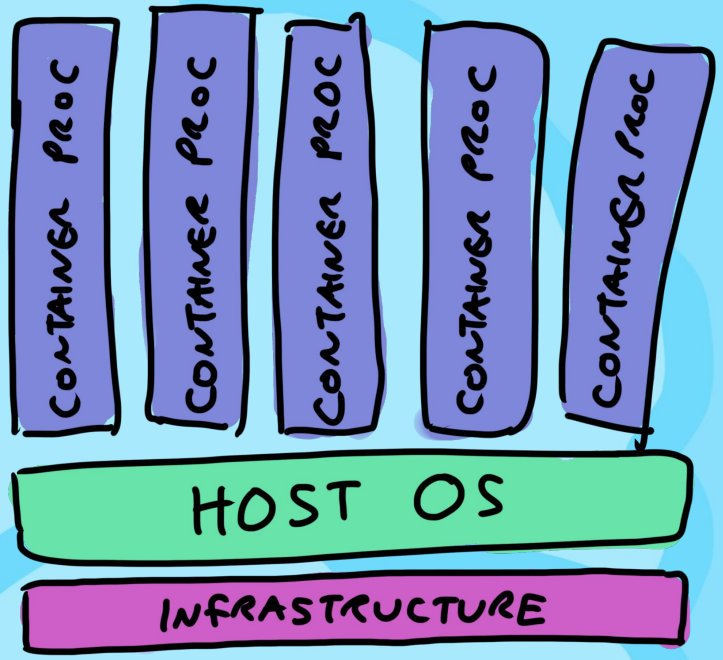
(As defined by the CNCF)

1. Open source
2. Microservices in containers
3. Dynamically orchestrated
4. Optimized resource utilization

Containers

Microservices in containers

- A unit of software
- A binary and all of its dependencies
- Containers share an OS
- Run as resource isolated processes



@jdrawlings

@bobcatwilson

Containers: the benefits

- Increased developer ease for building, packaging and running applications
- Fast startup times
- Savings in operational costs compared with running in VMs

Containers and CI/CD

- All of the dependencies you need are in the container
- All you need to do is run the container

Cloud Native: Containers + Kubernetes

Microservices in containers } **Images / Containers**

Dynamically orchestrated

Optimized resource utilization } **Kubernetes**

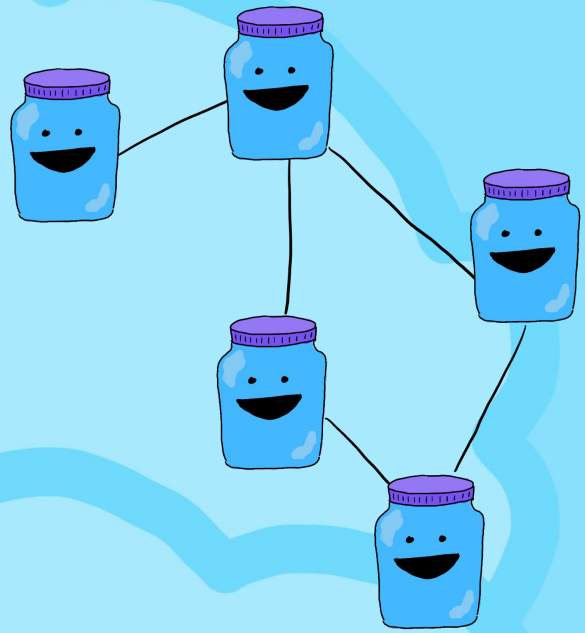
@jdrawlings

@bobcatwilson

Kubernetes

Dynamically orchestrated with optimized resource utilization

- Platform for managing containers
- Tell Kubernetes how to deploy your services and it does it
- Abstracts away the underlying hardware
  - Computing
  - Networking
  - Storage
- Cloud agnostic

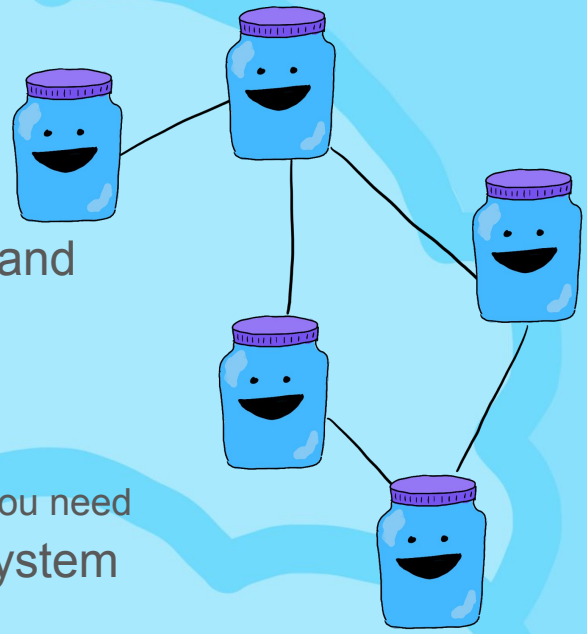


@jdrawlings

@bobcatwilson

Kubernetes: the benefits

- Standardisation
- Application portability
- Rich open source ecosystem with an innovative and vibrant community
- Better use of resources
  - Scaling up and down with demand
  - Serverless models mean you only pay for the compute you need
- Microservices translates naturally into a plugin system



@jdrawlings

@bobcatwilson

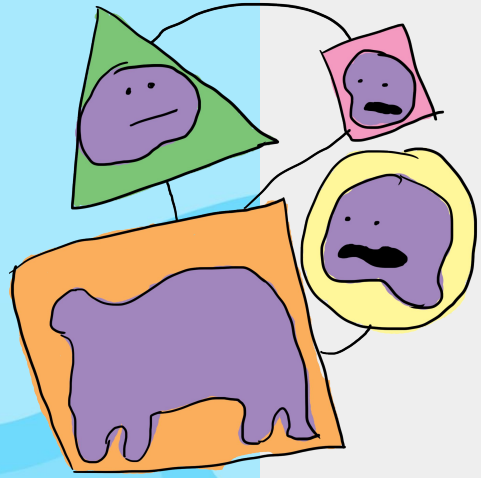
Kubernetes and CI/CD

Same ol’ Challenges

- Same old problems in a different form
  - Images instead of binaries
  - Clusters / Many environments

Kubernetes and CI/CD

New Challenges



- Some things more challenging
  - Microservices instead of monoliths

Jenkins X!
Tekton Pipelines!

@jdrawlings

@bobcatwilson

Jenkins

- History
  - Jenkins Server created in the form of Hudson in 2004
  - Almost 200,000 Jenkins Servers running *
  - 15,000,000 Jenkins users
- Present day challenges
  - Single Point of Failure
  - Large JVM requiring lots of memory and always running even if no builds required
  - Scaling jobs leads to issues as Pipelines are executed on the Jenkins Server

* Source https://stats.jenkins.io

@jdrawlings

@bobcatwilson

Jenkins X

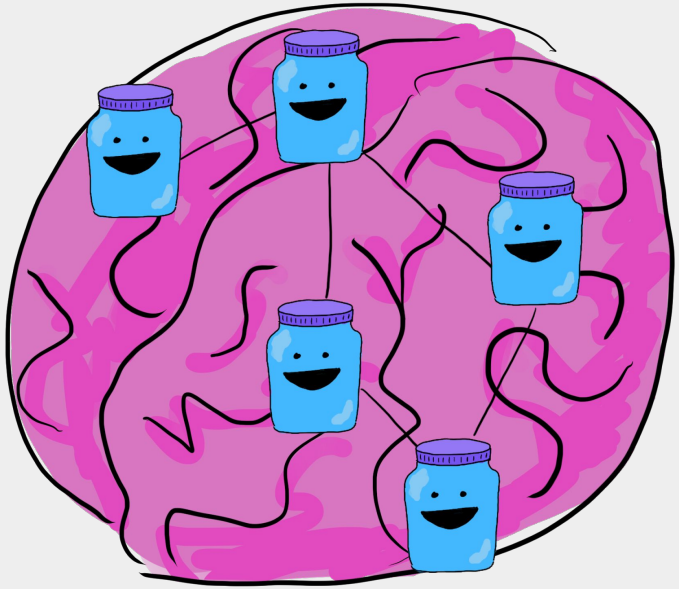
- Developer experience for Kubernetes
- Build traditional and modern cloud native workloads
- Create new or import existing applications onto Kubernetes
- Automated CI/CD
- Environments
- GitOps for environment promotion
- New extensibility model based on modern architectures
- Pluggable pipeline execution engines

@jdrawlings

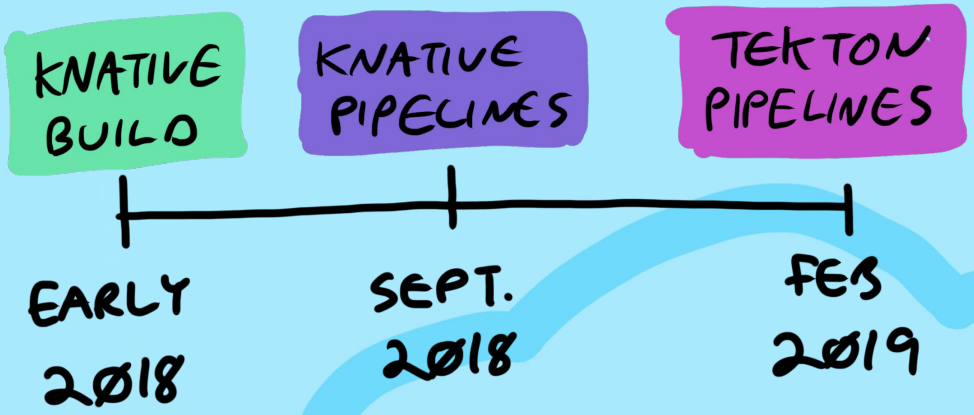
@bobcatwilson

What’s Tekton Pipelines?

The brains of CI/CD on kubernetes



Tekton Pipelines: The story



@jdrawlings

@bobcatwilson

Tekton Pipelines: Goals

- Portability: a CI/CD shared API spec!
- Declarative: types!
- Decoupled:
  - Run a Pipeline with your own resources!
  - Run pieces of a Pipeline (Tasks) on their own!
- Targeting many deployment targets
  - First class container support
  - Kubernetes
  - And beyond!

@jdrawlings

@bobcatwilson

Tekton Pipelines: Who

A collaborative effort!

Contributors from:

- Google
- Cloudbees
- Pivotal
- RedHat
- IBM
- … and more!

New contributor friendly!

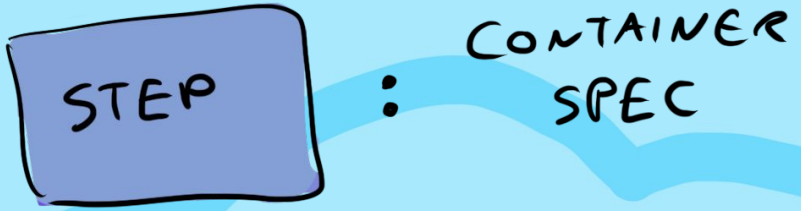
CRDs

Custom Resource Definitions

- Extending kubernetes with custom types
- Controllers act on Resources
- = CI/CD platform on Kubernetes

Steps

- Is actually a container spec (k8s type)
- Container image +
  - Environment variables
  - Arguments
  - Volumes
  - etc.



@jdrawlings

@bobcatwilson

Task CRD

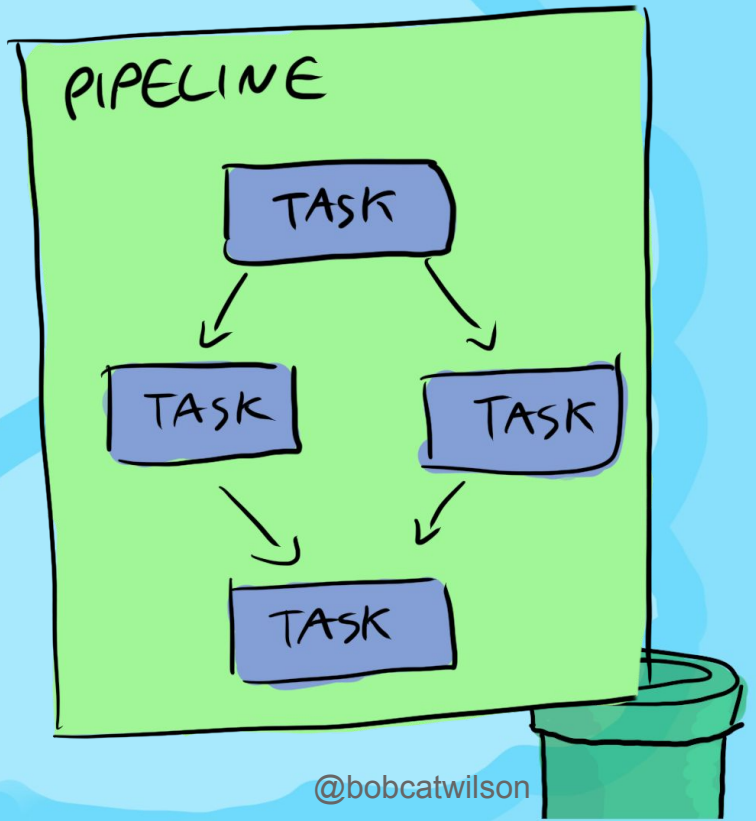
- New CRD
- Sequence of steps
- Run in sequential order
- Run on the same k8s node

@jdrawlings

@bobcatwilson

Pipeline CRD

- Express Tasks order
  - Sequentially
  - Concurrently
  - (Graph)
- Execute Tasks on different nodes
- Link inputs and outputs

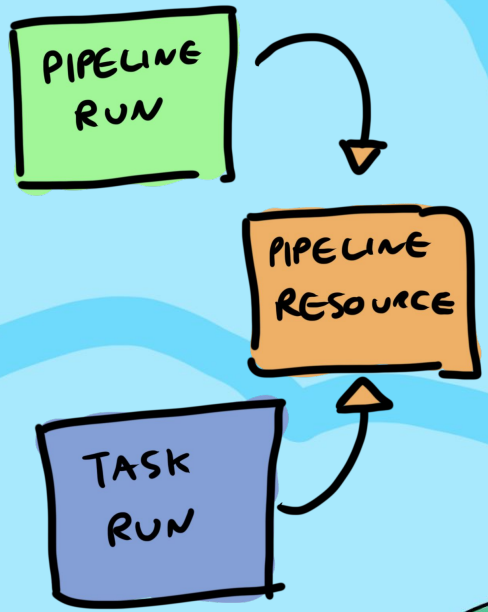


@jdrawlings

@bobcatwilson

Runtime CRDs

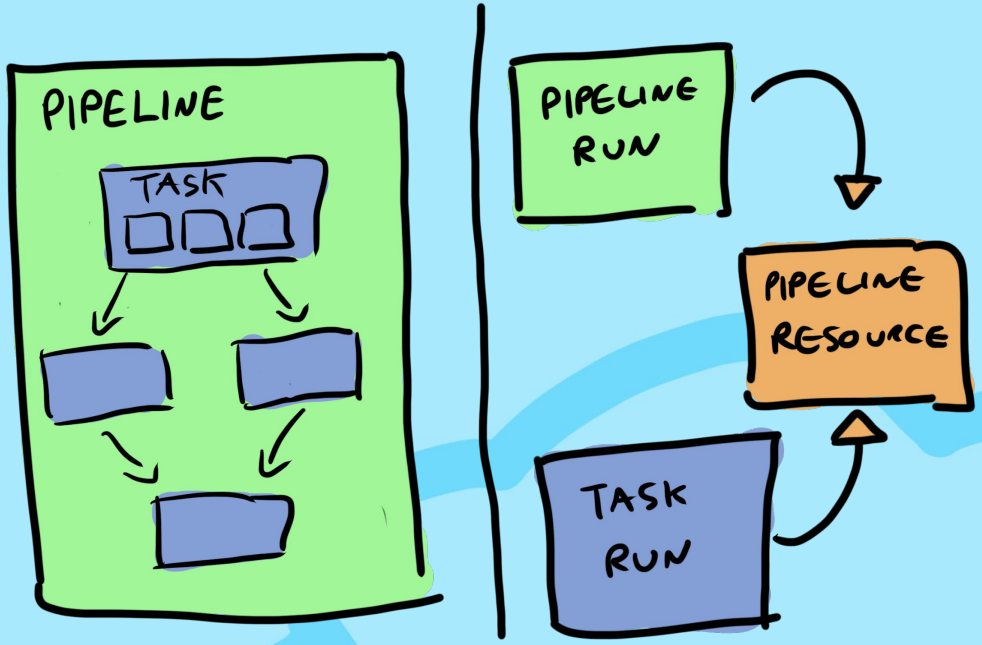
- Instances of Pipeline/Task:
  - PipelineRun
  - TaskRun
- PipelineResource
  - Runtime info like image registry, git repo, etc.



@jdrawlings

@bobcatwilson

Tekton Pipeline CRDs



@jdrawlings

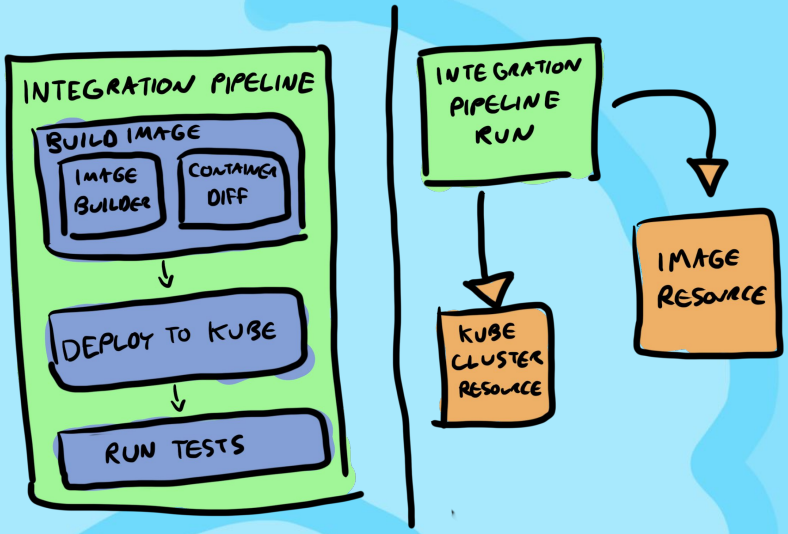
@bobcatwilson

Christie's earlier CI woes

1. Missing dependencies
2. Relying on production infrastructure:
    a. Image registry
    b. Kubernetes cluster
3. Didn't know any of this up front

Tekton Pipeline CRDs

- Pipeline:
  - Integration pipeline
- Tasks:
  - Build images
    - Step: Build image
    - Step: Run container-diff
  - Deploy to kube
  - Run tests
- PipelineRun
  - Would use my own image registry PipelineResource
  - Would use my own kubernetes cluster PipelineResource

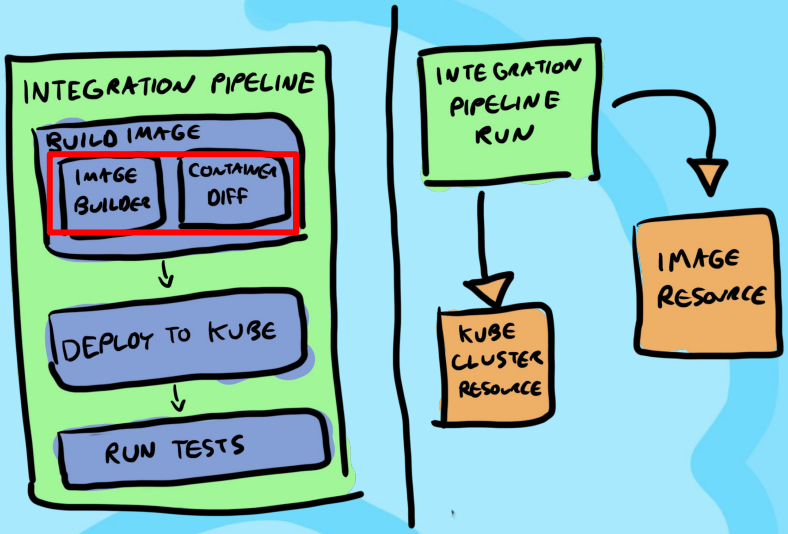


@jdrawlings

@bobcatwilson

Tekton Pipeline CRDs

- Pipeline:
  - Integration pipeline
- Tasks:
  - Build images
    - Step: Build image
    - Step: Run container-diff
  - Deploy to kube
  - Run tests
- PipelineRun
  - Would use my own image registry PipelineResource
  - Would use my own kubernetes cluster PipelineResource

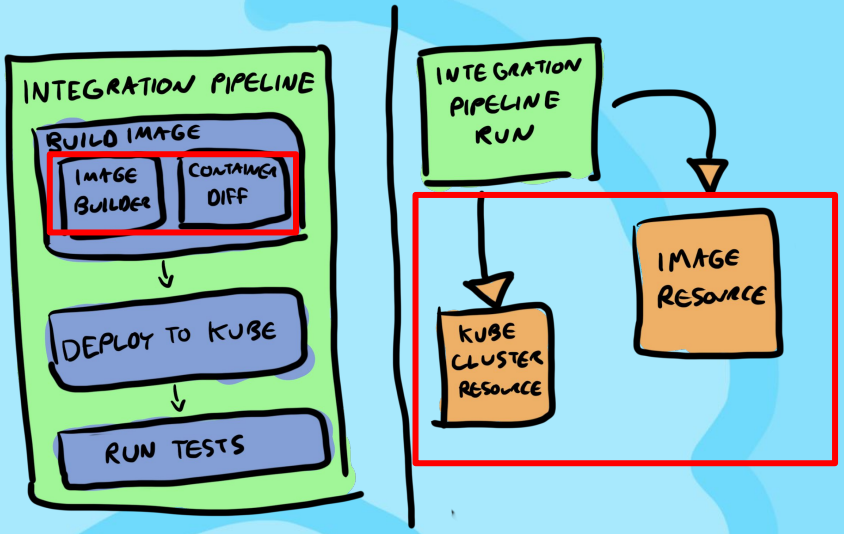


@jdrawlings

@bobcatwilson

Tekton Pipeline CRDs

- Pipeline:
  - Integration pipeline
- Tasks:
  - Build images
    - Step: Build image
    - Step: Run container-diff
  - Deploy to kube
  - Run tests
- PipelineRun
  - Would use my own image registry PipelineResource
  - Would use my own kubernetes cluster PipelineResource



@jdrawlings

@bobcatwilson

What do I need to run this Pipeline?

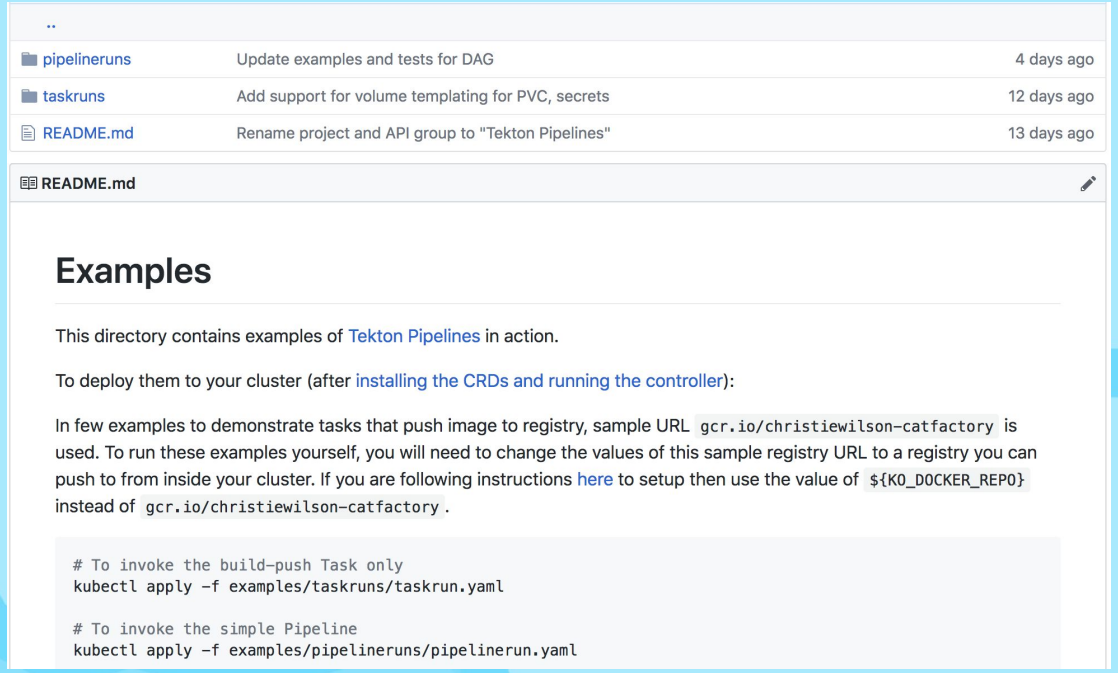
```yaml
kind: Pipeline
metadata:
  name: integration-pipeline
spec:
  resources:
  - name: source-repo
    type: git
  - name: app-image
    type: image
  - name: staging-cluster
    type: cluster
```

@jdrawlings

@bobcatwilson

Examples

github.com/knative/build-pipeline/tree/master/examples



..

pipelineruns	Update examples and tests for DAG	4 days ago
taskruns	Add support for volume templating for PVC, secrets	12 days ago
README.md	Rename project and API group to "Tekton Pipelines"	13 days ago

README.md

Examples

This directory contains examples of Tekton Pipelines in action.

To deploy them to your cluster (after installing the CRDs and running the controller):

In few examples to demonstrate tasks that push image to registry, sample URL `gcr.io/christiewilson-catfactory` is used. To run these examples yourself, you will need to change the values of this sample registry URL to a registry you can push to from inside your cluster. If you are following instructions here to setup then use the value of `${KO_DOCKER_REPO}` instead of `gcr.io/christiewilson-catfactory`.

```
# To invoke the build-push Task only
kubectl apply -f examples/taskruns/taskrun.yaml

# To invoke the simple Pipeline
kubectl apply -f examples/pipelineruns/pipelinerun.yaml
```

@jdrawlings

@bobcatwilson

Jenkins X + Pipelines

- This is an evolution of CI + CD using cloud capabilities
- Leverages Prow to trigger PipelineRuns
  - Prow is an event based git webhook handler
  - From the Kubernetes ecosystem
- Next Gen Pipeline jenkins-x.yml
- Dogfooding with Jenkins X has dramatically improved our builds example

@jdrawlings

@bobcatwilson

Demo!

@jdrawlings

@bobcatwilson

We can do better:
Cloud native CI/CD!

@jdrawlings

@bobcatwilson

Try it out!

- Jenkins X:
  - Quickstart: https://jenkins-x.io/getting-started/next-gen-pipeline/
  - Contribute: jenkins-x.io/contribute/
- Tekton Pipelines:
  - Quickstart: github.com/knative/build-pipeline/blob/master/docs/tutorial.md
  - Contributing guide: github.com/knative/build-pipeline/blob/master/CONTRIBUTING.md

@jdrawlings

@bobcatwilson

Faster!
Declarative!
Reproducible!
Cloud Native!

@jdrawlings

@bobcatwilson

Thanks!