



Tekton Pipelines As Code

An opinionated CI based on Tekton Pipelines

Shivam Mukhade

Software Engineer, Red Hat

Agenda

- CI/CD Pipeline?
- Tekton Overview
- Tekton Pipelines Demo
- Introduction to Pipelines As Code?
- Why Pipelines As Code?
- Demo :)

What is a CI/CD pipeline?

- A continuous integration and continuous deployment (CI/CD) pipeline is a series of steps that must be performed in order to deliver a new version of software.
Ref.
- By automating CI/CD throughout development, testing, production, and monitoring phases of the software development lifecycle, organizations are able to develop higher quality code, faster.
- Examples:
 - When someone create a pull request, run unit test, e2e, build test and deploy
 - When a commit is merged, deploy the code



An open-source project for providing a set of shared and standard components for building Kubernetes-style CI/CD systems



CD.FOUNDATION

Governed by the Continuous Delivery Foundation

Contributions from Google, Red Hat, Cloudbees, IBM, Pivotal and many more

Tekton Projects

Pipelines

Triggers

CLI

Dashboard

Operator

Catalog

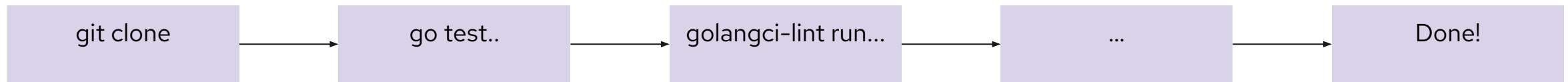
Hub

Chains

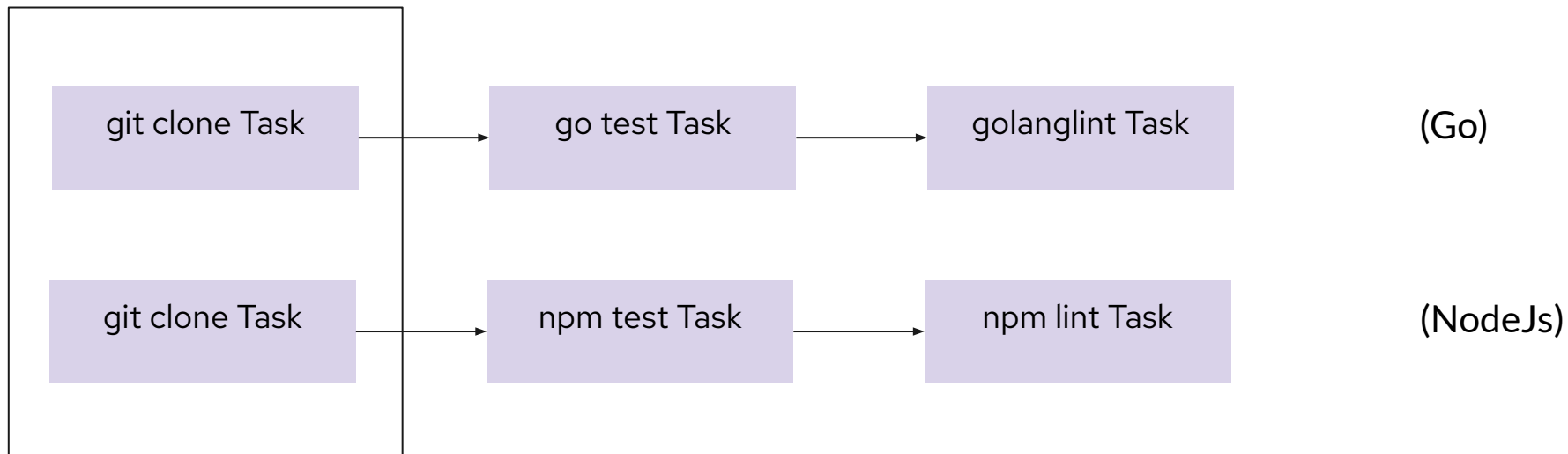
Results

Example

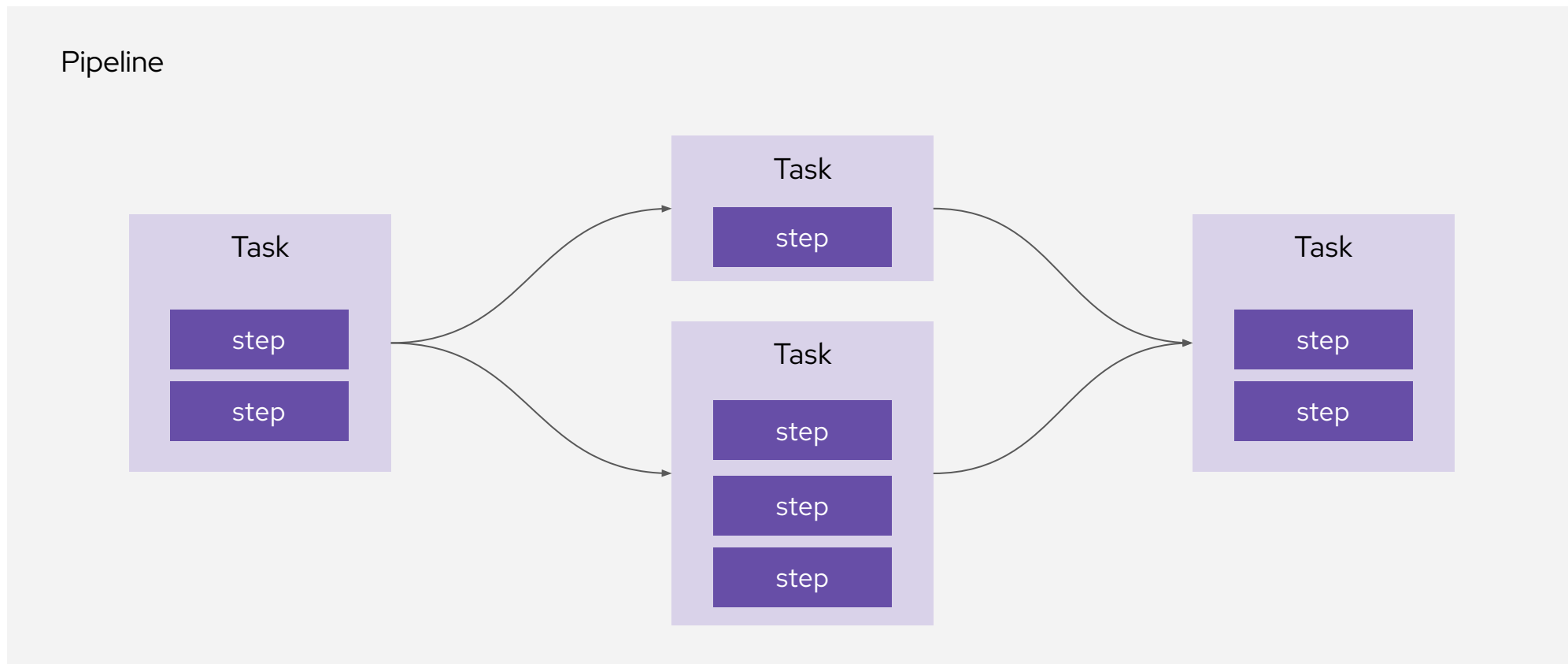
If we do it manually for a go project..



Now, let's see how to define using Tekton resources...



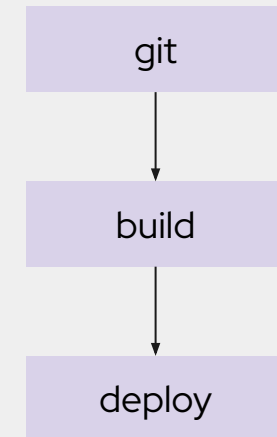
Tekton Concepts



Tekton Concepts: Pipeline

- A graph of Tasks: concurrent & sequential
- Tasks run on different nodes
- Task execution logic
 - Conditional
 - Retries
- Share data between tasks

```
kind: Pipeline
metadata:
  name: deploy-dev
spec:
  params:
    - name: IMAGE_TAG
  tasks:
    - name: git
      taskRef:
        name: git-clone
        params: [...]
    - name: build
      taskRef:
        name: buildah
        params: [...]
        runAfter: ["git"]
    - name: deploy
      taskRef:
        name: oc-deploy
        params: [...]
        runAfter: ["build"]
```



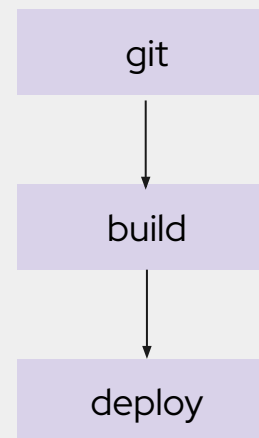
Task, Pipeline & PipelineRun

```
kind: Task
metadata:
  name: git-clone
spec:
  ...
```

```
kind: Task
metadata:
  name: buildah
spec:
  ...
```

```
kind: Task
metadata:
  name: oc-deploy
spec:
  ...
```

```
kind: Pipeline
metadata:
  name: deploy-dev
spec:
  params:
    - name: IMAGE_TAG
  tasks:
    - name: git
      taskRef:
        name: git-clone
      params: [...]
    - name: build
      taskRef:
        name: buildah
      params: [...]
      runAfter: ["git"]
    - name: deploy
      taskRef:
        name: oc-deploy
      params: [...]
      runAfter: ["build"]
```



```
kind: PipelineRun
metadata:
  name: deploy-dev-run-1
spec:
  pipelineRef:
    name: deploy-dev
```

```
kind: PipelineRun
metadata:
  name: deploy-dev-run-2
spec:
  pipelineRef:
    name: deploy-dev
```

```
kind: PipelineRun
metadata:
  name: deploy-dev-run-3
spec:
  pipelineRef:
    name: deploy-dev
```

Tekton Hub (hub.tekton.dev)

The screenshot displays the Tekton Hub homepage, which is a platform for discovering, searching, and sharing reusable Tekton Tasks and Pipelines. The interface features a dark header with the Tekton Hub logo, a search bar, and a 'Login' button. Below the header, a 'Welcome to Tekton Hub' message is followed by the tagline 'Discover, search and share reusable Tasks and Pipelines'.

The main content area is a grid of task cards, each representing a reusable task. Each card includes a task icon, a star rating, the task name, version, a brief description, and a 'Updated' timestamp. The tasks are categorized by 'Kind' (Pipeline or Task) and 'Platform' (Linux/amd64, Linux/ppc64le, Linux/s390x, Linux/arm64). A sidebar on the left provides filters for 'Catalog' (Tekton) and 'Category' (Automation, Build Tools, CI, Cloud, Code Quality, Continuous Integration, Deployment, Developer Tools, Git, Image Build, Integration & Delivery, Kubernetes, Messaging, Monitoring, Networking, Openshift, Publishing, Security, Storage, Testing).

The tasks displayed in the grid include:

- Buildpacks** (v0.2): The Buildpacks pipeline builds source from a Git repository into a container image and pushes it to a registry, using Cloud Native Buildpacks. Updated 2 months ago.
- git cli** (v0.4): This task can be used to perform git operations. Git command that needs to be run can be passed as a script to the task. This task needs authentication to git in order to push after the... Updated a month ago.
- golang build** (v0.3): This Task is Golang task to build Go projects. Updated 8 months ago.
- jenkins operation** (v0.1): The following task can be used to interact with the the Jenkins REST API. Updated 10 months ago.
- pull request** (v0.1): This Task allows a user to interact with an SCM (source control management) system through an abstracted interface This Task works with both public SCM instances and self-. Updated a year ago.
- RedHat CodeReady Dependency Analysis** (v0.1): The RedHat CodeReady Dependency Analysis task is an interface between Tekton and Red Hat CodeReady Dependency Analytics. Updated 10 months ago.
- ssh remote commands** (v0.1): The following task can be used to execute commands on remote machine. The following task takes host and required credentials as input along with the script and execute. Updated 10 months ago.
- rsync** (v0.1): This task can be used to synchronize local and remote files. Updated a year ago.
- send-to-telegram** (v0.1): These tasks post a simple message to a telegram chat. This task uses the Bot API of telegram to send a message. Updated a year ago.
- trigger jenkins job** (v0.1): The following task can be used to trigger a Jenkins job using CURL request from a Tekton Task. Updated a year ago.
- YQ replace** (v0.3): This task can be used to replace fields in YAML files. For example for altering helm charts on GitOps repos. Updated 7 months ago.
- Buildpacks** (v0.5): The Buildpacks task builds source into a container image and pushes it to a registry, using Cloud Native Buildpacks. Updated a month ago.
- Buildpacks (phases)** (v0.2): The Buildpacks-Phases task builds source into a container image and pushes it to a registry, using Cloud Native Buildpacks. This task separately calls the aspects of the Cloud. Updated 10 months ago.
- helm-upgrade-from-source** (v0.3): These tasks will install / upgrade a helm chart into your Kubernetes / OpenShift Cluster using Helm. Updated a year ago.
- Build and upload container image using Kaniko** (v0.6): This Task builds a simple Dockerfile with kaniko and... Updated a year ago.
- buildah** (v0.5): Buildah task builds source into a container image and then pushes it to a container registry. Buildah Task builds source into a container image using Project. Updated a year ago.
- jib maven** (v0.4): This Task builds Java/Kotlin /Groovy/Scala source into a container image using Google's Jib tool. Jib works with Maven and Gradle projects, and this. Updated a year ago.
- Ansible Runner** (v0.2): Task to run Ansible playbooks using Ansible Runner. Updated 7 months ago.
- Amazon ECR Login** (v0.1): This task retrieves an 'authentication token' using the GetAuthorizationToken API that you can use to authenticate to an 'Amazon ECR registry'. You can. Updated a year ago.
- curl** (v0.1): This task performs curl operation to transfer data from internet. Updated a year ago.
- git batch merge** (v0.2): This task takes a set of refsspecs, fetches them and performs git operations (cherry-pick or merge) to apply them in order on the given base revision (default. Updated a year ago.

Tekton Triggers

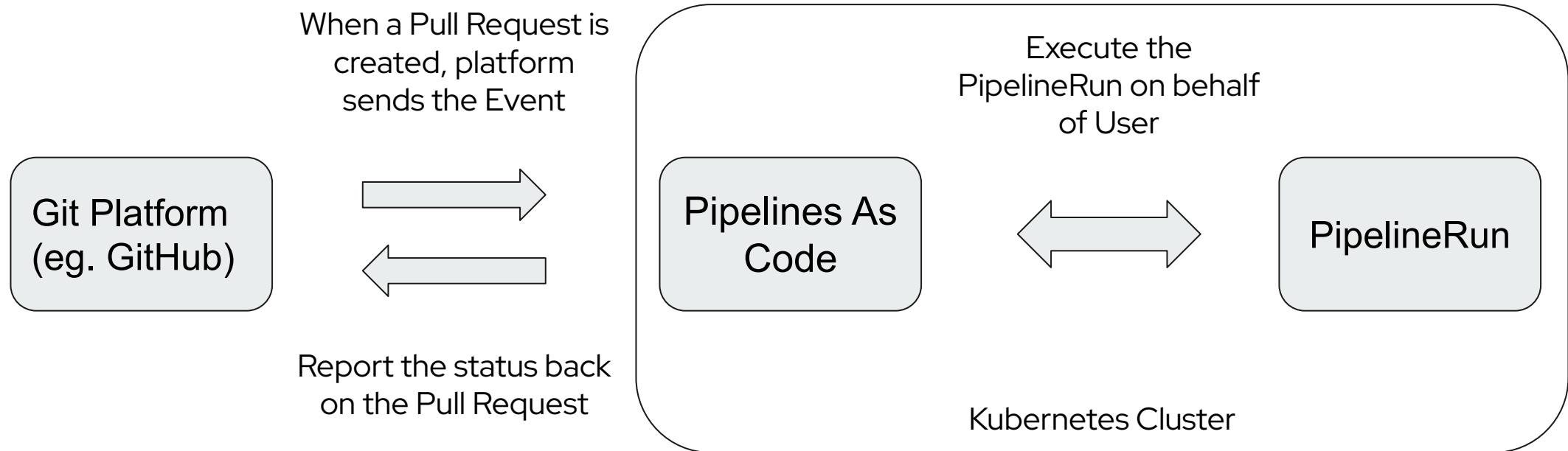
- Event triggering with Tekton!
- Triggers will execute pipeline but won't report





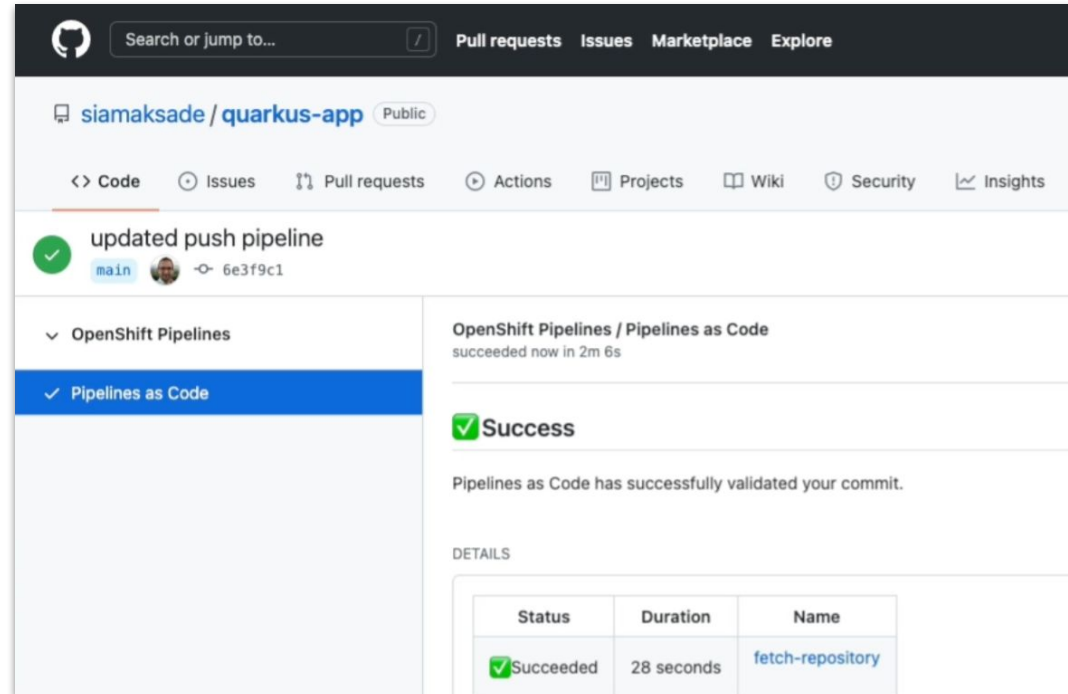
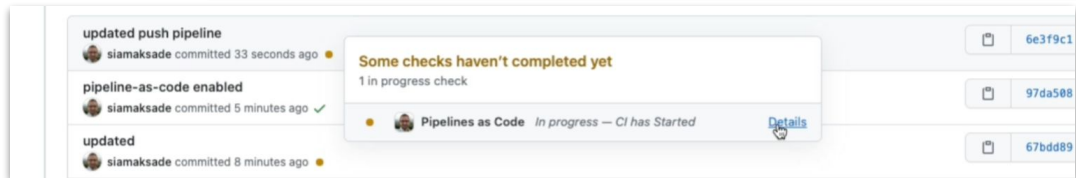
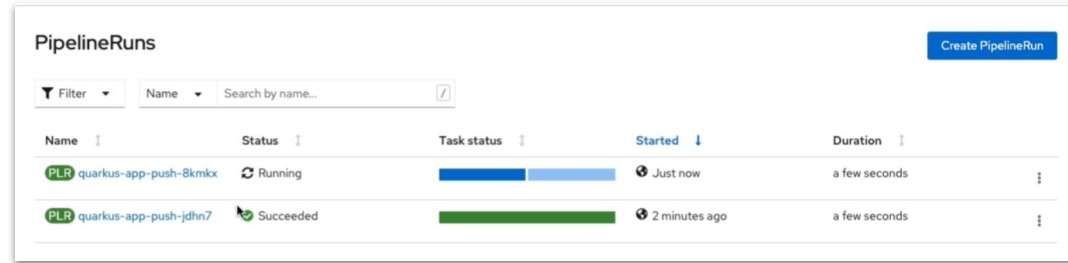
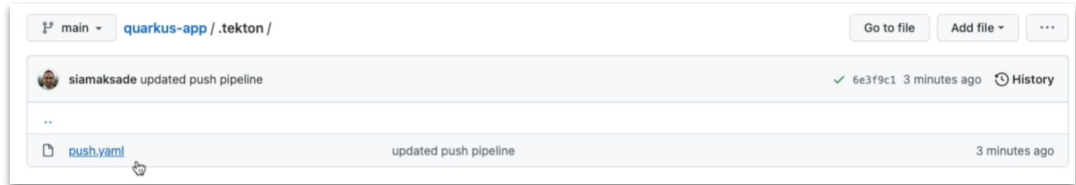
Pipelines As Code

Overview



Pipelines as Code

- Git as the source of pipeline definition
- Pipelines retrieved and run from Git
- Pipeline per Git event (push, pull-request)
- Pipeline status on GitHub commit and pr
- Integrated with GitHub Checks
- Restart pipelines via pull-request comments



Pipelines as Code

- Support using
 - GitHub App
 - Webhook
 - GitHub
 - GitLab
 - Bitbucket Cloud/Server

Conversation 0 Commits 8 Checks 4 Files changed 3

✓ dont-fail 44d1373

sm-pac-demo

- ✓ Pipelines as Code CI / pac-app-...
- ✓ Pipelines as Code CI / pac-app-...
- > Deleted GitHub App

sm-pac-demo / Pipelines as Code CI / pac-app-pull-request-1
succeeded 8 days ago in 1m 3s


Success





Pipelines as Code CI/pac-app-pull-request-1 has **successfully** validated your commit.

DETAILS

Status	Duration	Name
✓ Succeeded	36 seconds	fetch-repository
✓ Succeeded	26 seconds	noop-task

[View more details on sm-pac-demo](#)

 **Some checks were not successful** [Hide all checks](#)
2 failing and 2 successful checks

✗	 Pipelines as Code CI / pac-app-pull-request	Failing after 1m — Failed	Details
✗	 Pipelines as Code CI / pac-app-pull-request-1	Failing after 1m — Failed	Details
✓	 Pipelines as Code CI / pac-app-pull-request	Successful in 1m — Success	Details
✓	 Pipelines as Code CI / pac-app-pull-request-1	Successful in 1m — Success	Details

✓ **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request or view [command line instructions](#).

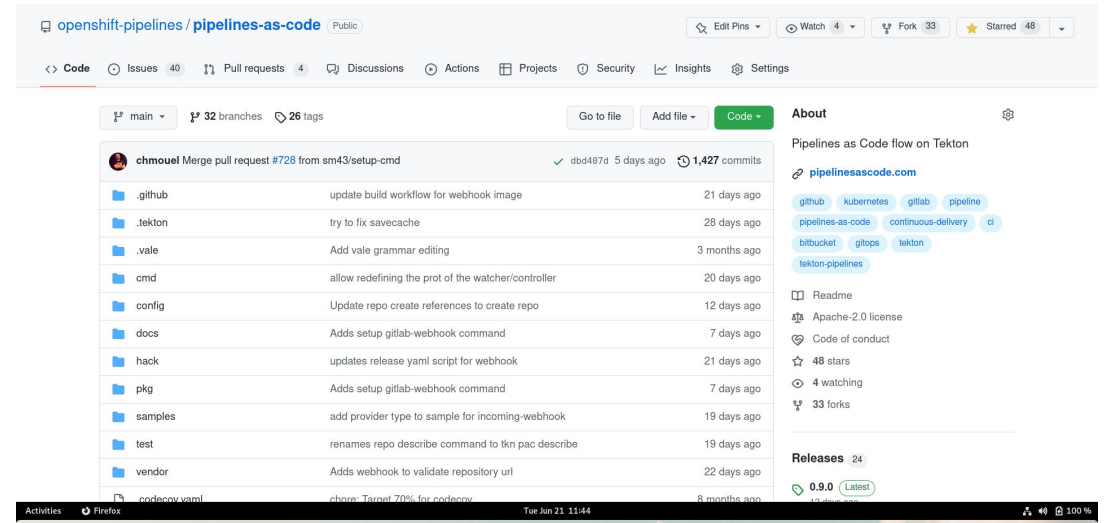
sm43 commented on 11 May

Pipelines as Code CI/pac-webhook-pull-request has **failed**.

Status	Duration	Name
✓ Succeeded	9 seconds	fetch-repository
✗ Failed	6 seconds	noop-task

What's Next?

- [openshift-pipelines/pipelines-as-code](https://github.com/openshift-pipelines/pipelines-as-code)
- <https://pipelinesascode.com/>



Pipelines as Code

Search

- Installation
 - Overview
 - Installation
 - Settings
- GitHub Apps
- GitHub Webhook
- Gitlab
- Bitbucket Cloud
- Bitbucket Server
- Kubernetes
- Usage Guide
- Repository CRD
- Resolver
- Authoring PipelineRun
- Running the PipelineRun
- PipelineRun status
- Private Repositories
- PipelineRuns Cleanup
- CLI tkn-pac
- Incoming Webhook
- PAC Flow Diagram
- GitHub Pipelines as Code
- TektonCD Documentation

Pipelines as Code

An opinionated CI based on OpenShift Pipelines / Tekton.

Introduction

Pipelines as Code let you use the [Pipelines as Code](#) flow directly with OpenShift Pipelines.

The goal of Pipelines as Code is to let you define your [Tekton](#) templates inside your source code repository and have the pipeline run and report the status of the execution when triggered by a [Pull Request](#) or a [Push](#).

Features

- Pull-request status support: When iterating over a Pull Request, Statuses and Control is done on GitHub.
- GitHub Checks API support to set the status of a PipelineRun including rechecks
- GitHub Pull Request and Push event support
- Pull-request "[GitOps](#)" actions through comments with `/retest`, `/test <pipeline-name>` and so on.
- Automatic Task resolution in Pipelines (local Tasks, Tekton Hub, and remote URLs)
- Efficient use of GitHub blobs and objects API for retrieving configurations
- Git events Filtering and support for separate pipelines for each event
- Gitlab, Bitbucket Server, Bitbucket Cloud and GitHub Webhook support.
- `tkn-pac` plug-in for Tekton CLI for managing pipelines-as-code repositories and bootstrapping.

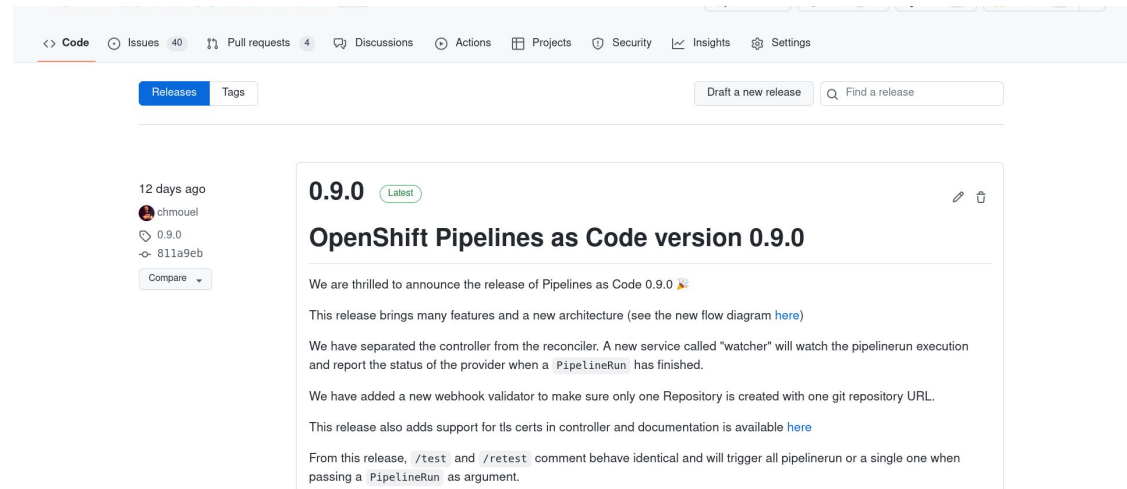
Getting Started

The easiest way to get started is to use the `tkn pac` CLI and its `bootstrap` command.

Start downloading and install the tkn-pac CLI following [these instructions](#) and while connected to your cluster (for example using `kind` for testing) run the command :

```
-$ tkn pac bootstrap
```

and follow the questions to get Pipelines as Code installed on your cluster. It will then help you create a GitHub Application to connect




References

- [Pipelines As Code](#)
- <https://pipelinesascode.com/>

Blogs: (sm43.medium.com)

- [World of Tekton](#)
- [OpenShift/Tekton Pipelines As Code](#)

 Pipelines as Code

Installation

Overview

Installation

Settings

GitHub Apps

GitHub Webhook

Gitlab

Bitbucket Cloud

Bitbucket Server

Kubernetes

Usage Guide

Repository CRD

Resolver

Authoring PipelineRun

Running the PipelineRun

PipelineRun status

Private Repositories

PipelineRuns Cleanup

CLI tkn-pac

Incoming Webhook

PAC Flow Diagram

GitHub Pipelines as Code

TektonCD Documentation

Pipelines as Code

An opinionated CI based on OpenShift Pipelines / Tekton.

Introduction

Pipelines as Code let you use the [Pipelines as Code flow](#) directly with OpenShift Pipelines.

The goal of Pipelines as Code is to let you define your [Tekton](#) templates inside your source code repository and have the pipeline run and report the status of the execution when triggered by a `Pull Request` or a `Push`.

Features

- Pull-request status support: When iterating over a Pull Request. Statuses and Control is done on GitHub.
- GitHub Checks API support to set the status of a PipelineRun including rechecks
- GitHub Pull Request and Push event support
- Pull-request “*GitOps*” actions through comments with `/retest`, `/test <pipeline-name>` and so on.
- Automatic Task resolution in Pipelines (local Tasks, Tekton Hub, and remote URLs)
- Efficient use of GitHub blobs and objects API for retrieving configurations
- Git events Filtering and support for separate pipelines for each event
- Gitlab, Bitbucket Server, Bitbucket Cloud and GitHub Webhook support.
- `tkn-pac` plug-in for Tekton CLI for managing pipelines-as-code repositories and bootstrapping.

Getting Started

The easiest way to get started is to use the `tkn pac` CLI and its `bootstrap` command.

Start downloading and install the tkn-pac CLI following [these instructions](#) and while connected to your cluster (for example using `kind` for testing) run the command :

```
$ tkn pac bootstrap
```

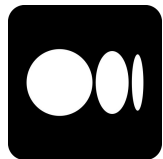
Thank you



sm43



@smeeee43



sm43



sm43 (k8s / tektoncd)

