

Next Generation CI/CD mit OpenShift Pipelines, Projekt Tekton und GitOps

Daniel Brintzinger
Senior Solution Architect
Red Hat

Sebastian Tiggelcamp
DevOps Engineer
Consol

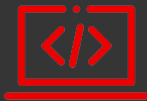
Ortwin Schneider
Solution Architect
Red Hat



What we'll discuss today



Introduction to Project Tekton,
OpenShift Pipelines & GitOps

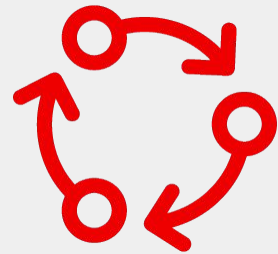


Live Demo



Lessons Learned and Roadmap

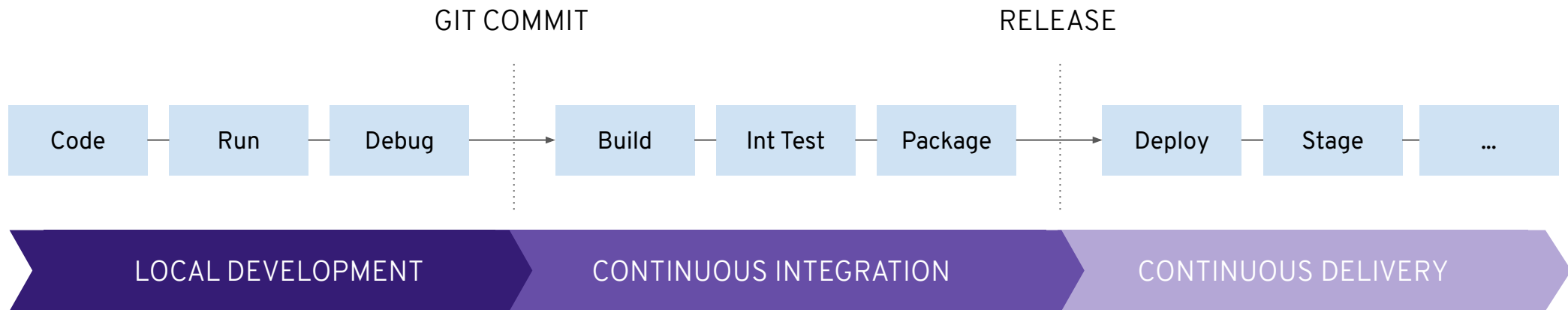
Introduction to Project Tekton, OpenShift Pipelines & GitOps



What is CI/CD?



Continuous Integration and Continuous Delivery (CI/CD)



Kubernetes



Kubernetes



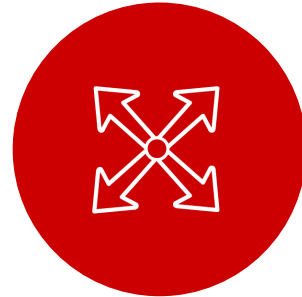
Kubernetes

What is Cloud-Native CI/CD?



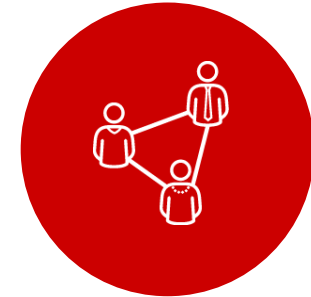
Containers

Built for container apps and runs on Kubernetes



Serverless

Runs serverless with no CI/CD engine to manage and maintain





DevOps

Designed with microservices and distributed teams in mind

Why Cloud-Native CI/CD?

Traditional CI/CD	Cloud-Native CI/CD
Designed for Virtual Machines	Designed for Containers and Kubernetes
Require IT Ops for CI engine maintenance	Pipeline as a service with no Ops overhead
Plugins shared across CI engine	Pipelines fully isolated from each other
Plugin dependencies with undefined update cycles	Everything lifecycle'd as container images
No interoperability with Kubernetes resources	Native Kubernetes resources
Admin manages persistence	Platform manages persistence
Config baked into CI engine container	Configured via Kubernetes ConfigMaps

Why Cloud-Native CI/CD?

Traditional CI/CD	Cloud-Native CI/CD
Designed for Virtual Machines	Designed for Containers and Kubernetes
Require IT Ops for CI engine maintenance	Pipeline as a service with no Ops overhead
 Jenkins Plugins shared across CI engine Plugin dependencies with undefined update cycles	 TEKTON Pipelines fully isolated from each other Evaluating lines of code in containers
No interoperability with Kubernetes resources	Native Kubernetes resources
Admin manages persistence	Platform manages persistence
Config baked into CI engine container	Configured via Kubernetes ConfigMaps

OpenShift Pipelines



OpenShift Pipelines



Kubernetes-native
declarative
Pipelines with
Tekton



Serverless CI/CD
with no single server
to share and
maintain



Run pipelines in
isolated containers with
all required
dependencies



Standard and
portable to any
Kubernetes
platform



Web, CLI, and
Visual Studio
Code and IDE
plugins

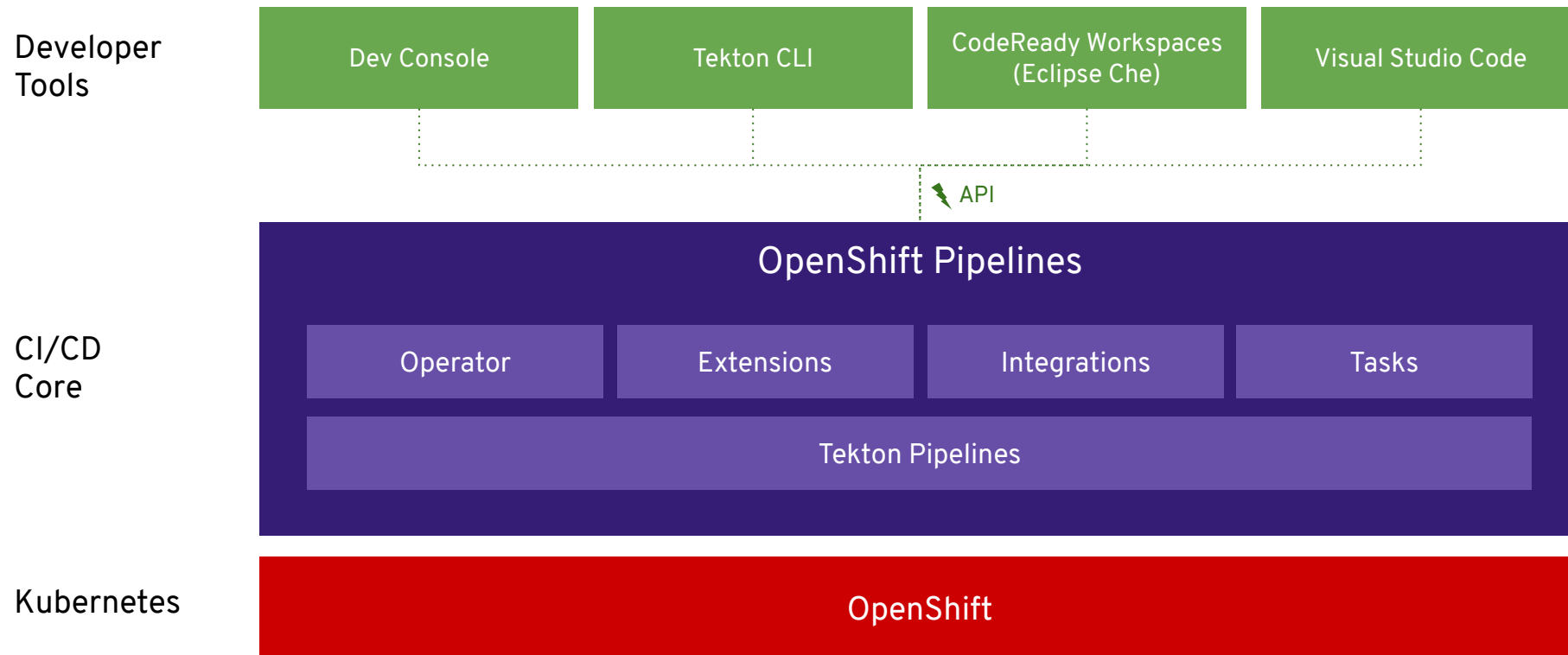


An open-source project for providing a set of shared and standard components for building Kubernetes-style CI/CD systems



Governed by the Continuous Delivery Foundation
Contributions from Google, Red Hat, Cloudbees, IBM, Pivotal and many more

OpenShift Pipelines Architecture



Tekton Concepts

Step

Run commands in a container with volumes, env vars, etc

Task

A list of steps that run sequentially in the same pod

Pipeline

A graph of tasks executed in a certain order

Pipeline Resource

Inputs and outputs to tasks and pipelines (git, image, etc)

Task Run

An invocation of a task with inputs and outputs

Pipeline Run

An invocation of a pipeline with inputs and outputs

Condition

An check that can determine if a task should be executed

Catalog

An collection of reusable tasks

Triggers

A Tekton sub-project to start pipelines based on events

Steps

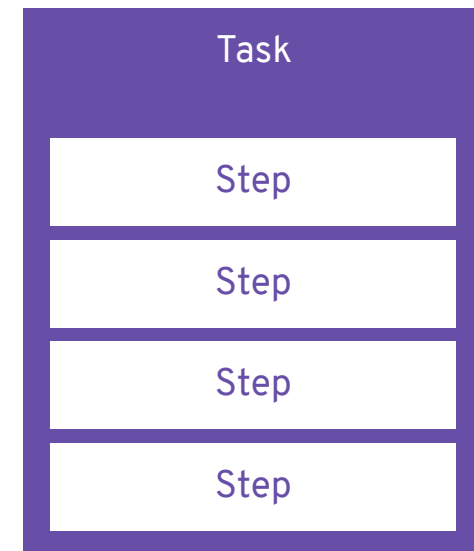
- Run command or script in a container
- Kubernetes container spec
 - Env vars
 - Volumes
 - Config maps
 - Secrets

```
- name: build
  image: maven:3.6.0-jdk-8-slim
  command: ["mvn"]
  args: ["install"]
```

```
- name: parse-yaml
  image: python3
  script: |-
    #!/usr/bin/env python3
    ...
```

Task

- Defines a unit of work to be executed
- A list of steps to run sequentially
- Step containers run in the task pod
- Has inputs, outputs and parameters
- Workspaces and results for sharing data
- Can run independent of pipelines



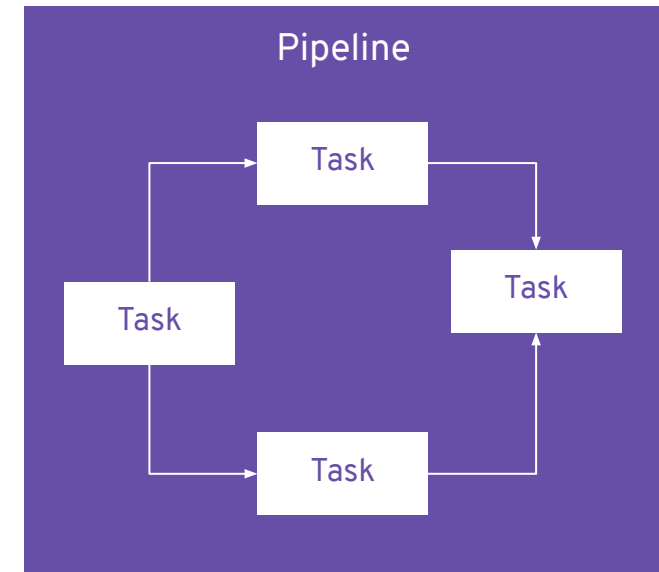
Example Tasks: Maven Install, AWS CLI, Kubectl Deploy, Security Scan, etc

Maven Task Example

```
kind: Task
metadata:
name: maven
spec:
  params:
    - name: goal
      type: string
      default: package
  steps:
    - name: mvn
      image: maven:3.6.0-jdk-8-slim
      command: [ mvn ]
      args: [ $(params.goal) ]
```


Pipeline

- Define Tasks execution order (graph)
- Inputs and parameters
- Retries tasks
- Conditional task execution
- Workspaces for sharing data between tasks
- Reusable across projects



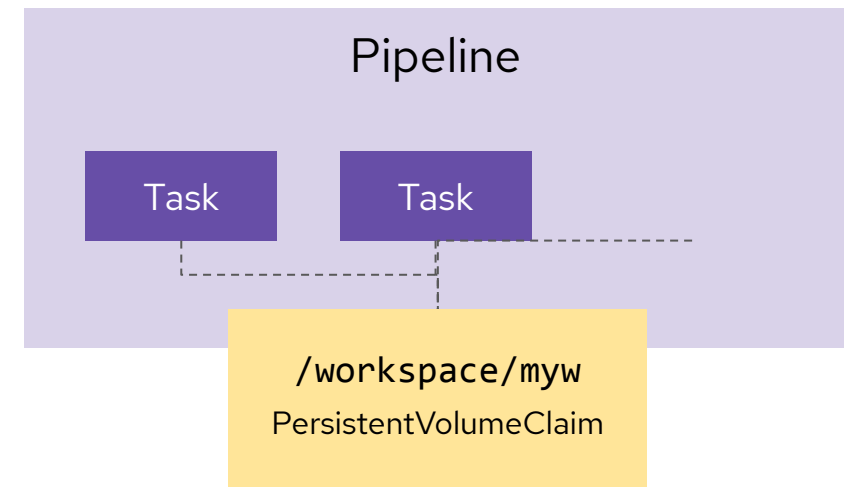
Sharing Data Between Tasks

Task: results

- Task exposes data as variables
- Suitable for small pieces of data
- Examples: commit id and branch name

Task: workspaces

- Shared volumes between tasks
 - Persistent volumes
 - Config maps
 - Secrets
- Suitable for large data
- Examples: code, binaries, reports



Conditions

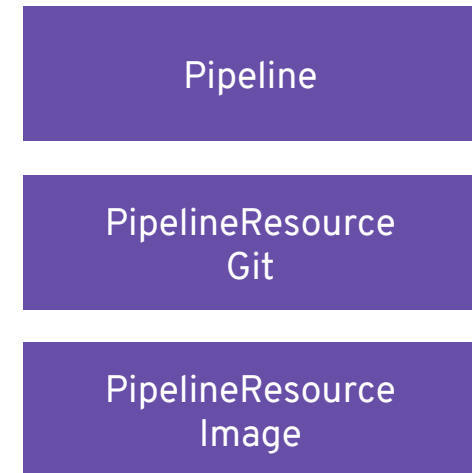
- Defines a single checks
- Used in conditional tasks
- Parameterized
- Command run in container
 - True: exit code - 0
 - False: non-zero exit code

```
kind: Condition
metadata:
name: deployment-exists
spec:
  params:
  - name: appName
  check:
    image: openshift-cli
    script: oc get deployment ${params.app}
```

```
kind: Pipeline
spec:
  tasks:
  - name: run-tests
    taskRef: { image: ui-test-runner }
    conditions:
    - conditionRef: deployment-exists
      params:
      - {name: appName, value: api }
```

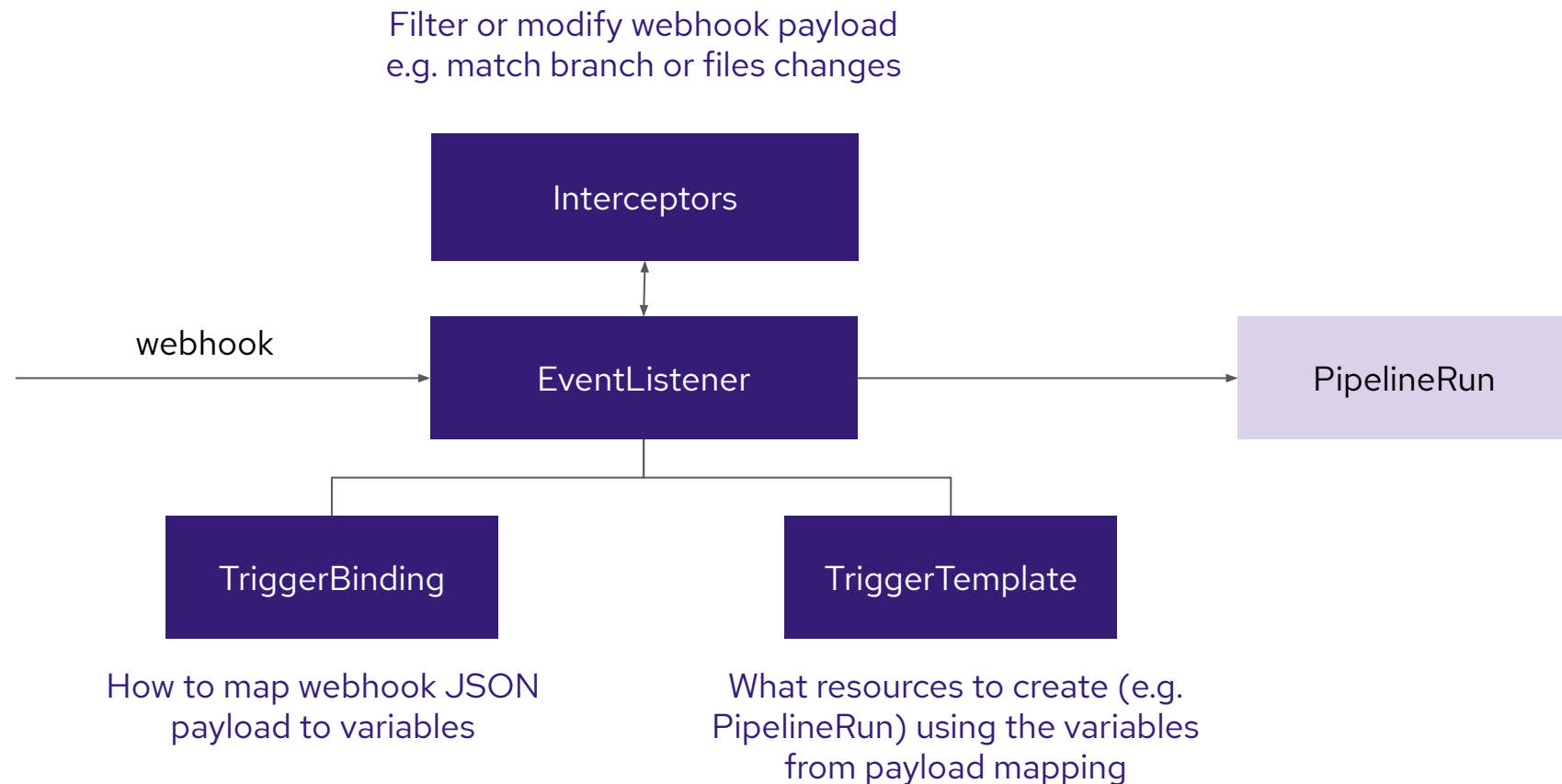
PipelineResource

- Inputs and outputs of tasks and pipelines
 - git repository
 - image in a registry
 - cluster credentials
 - storage
 - ...and mo
- Decoupled from pipeline definition
- Reusable across pipelines



Triggers

Run pipelines based on events like HTTP webhooks on commit, pull request, etc



Task Catalogs

A growing collection of reusable tasks

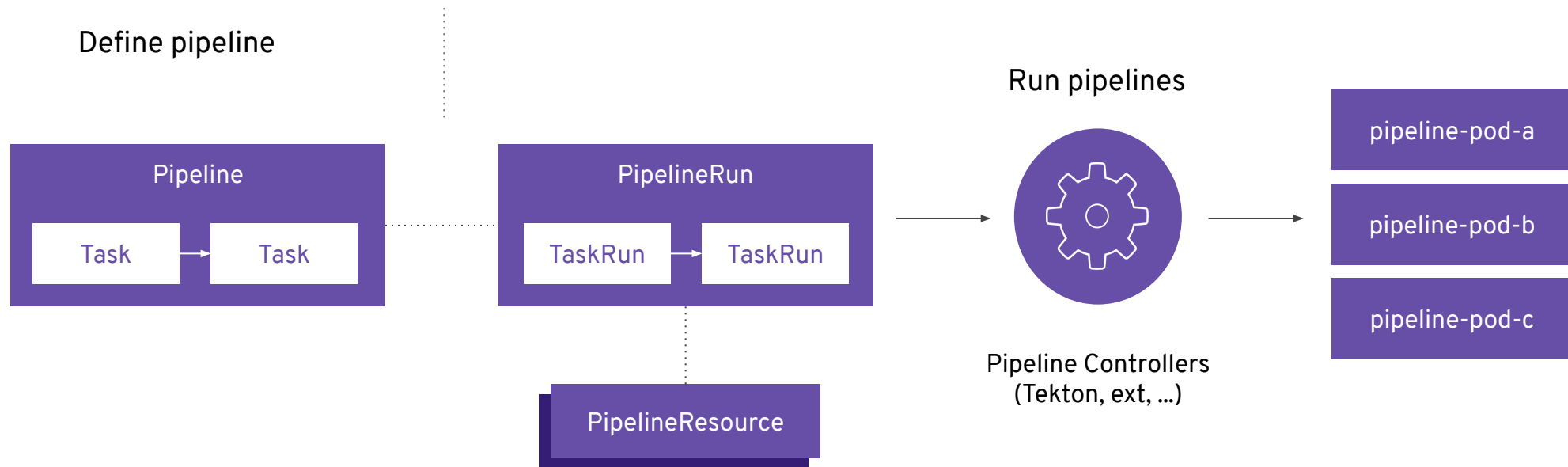
tektoncd / catalog

 .github	 kaniko
 argocd	 kn
 buildah	 knctl
 buildkit	 kubeval
 buildpacks	 makisu
 conftest	 openshift-client
 golang	 openwhisk
 jib-maven	 s2i

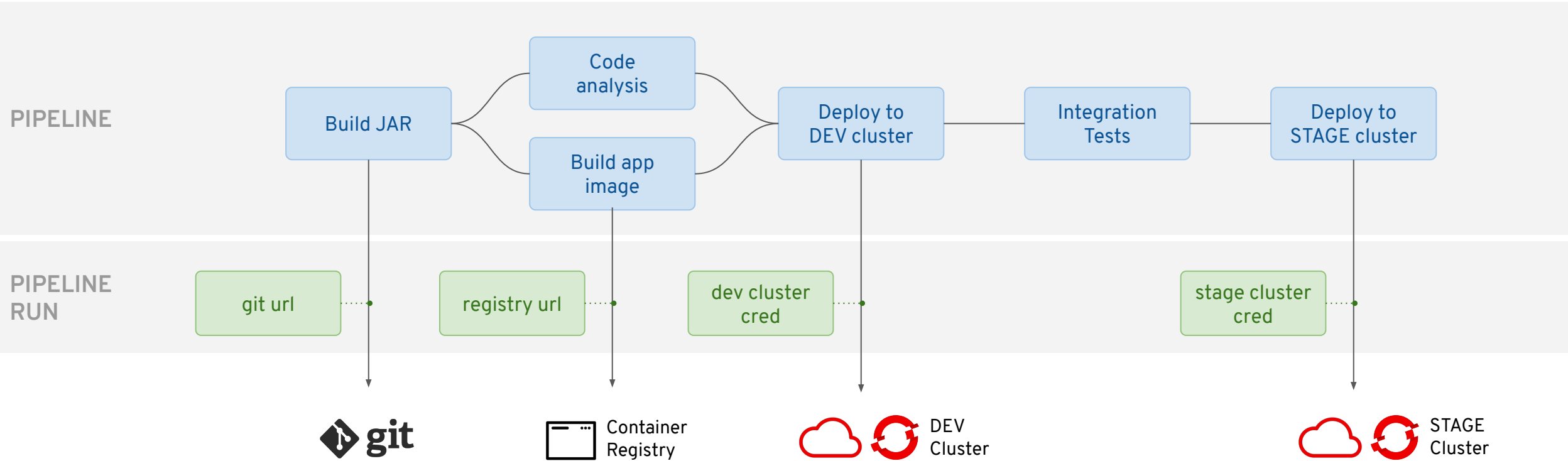
openshift / pipelines-catalog

 s2i-dotnet-1	 s2i-python-2
 s2i-dotnet-2	 s2i-python-3
 s2i-go	 s2i-ruby
 s2i-java-11	
 s2i-java-8	
 s2i-nodejs	
 s2i-perl	
 s2i-php	

OpenShift Pipelines Architecture



Tekton Pipeline Example



Install OpenShift Pipelines Operator

The screenshot shows the OpenShift OperatorHub interface. The left sidebar contains navigation options: Administrator, Home, Overview, Projects, Search, Explore, Events, Operators (selected), Installed Operators, Workloads, Serverless, Networking, Storage, Builds, Pipelines, and Monitoring. The main content area displays the details for the OpenShift Pipelines Operator (version 0.10.7, provided by Red Hat). The operator is currently installed, as indicated by the 'Installed Operator' status and the 'Uninstall' button. The interface also shows the operator's version, capability level (Basic Install), provider type (Community), provider (Red Hat), repository (https://github.com/openshift/tektoncd-pipeline-operator), and container image (quay.io/openshift-pipeline/openshift-pipelines-operator:v0.10.7). A list of features is provided, including standard CI/CD pipeline definition, build image support, multi-platform deployment, and integration with OpenShift Developer Console.

OpenShift Pipelines Operator
0.10.7 provided by Red Hat

[Uninstall](#)

Operator Version
0.10.7

Capability Level

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

Provider Type
Community

Provider
Red Hat

Repository
https://github.com/openshift/tektoncd-pipeline-operator

Container Image
quay.io/openshift-pipeline/openshift-pipelines-operator:v0.10.7

Installed Operator
This Operator has been installed on the cluster. [View it here.](#)

Features

- Standard CI/CD pipelines definition
- Build images with Kubernetes tools such as S2I, Buildah, Buildpacks, Kaniko, etc
- Deploy applications to multiple platforms such as Kubernetes, serverless and VMs
- Easy to extend and integrate with existing tools
- Scale pipelines on-demand
- Portable across any Kubernetes platform
- Designed for microservices and decentralised team
- Integrated with OpenShift Developer Console

Installation
OpenShift Pipelines Operator gets installed into a single namespace which would then install *OpenShift Pipelines* into the same namespace. *OpenShift Pipelines* is however cluster-wide and can run pipelines created in any namespace.

Red Hat
OpenShift Container Platform

Project: Project01

Pipelines > Pipeline Run Details Tech Preview

PR pipelinerun01a Running Actions

[Overview](#) [YAML](#) [Logs](#)

Pipeline Run Overview

```
graph LR; A[code compile] --> B[compile & test]; B --> C[unit test]; B --> D[security check]; C --> E[image build]; D --> E;
```

Name
pipelinerun01a

Namespace
NS project01

Labels
`app=dummy-mongo-pod-test` `bap.me/environment=dev`
`bap.me/track=experimental` `bap.me/tier=backend`

Annotations
[0 Annotations](#)

Created At
Aug 8, 4:00 pm

Red Hat
OpenShift Container Platform

Project: Project01

Pipelines > Pipeline Run Details Tech Preview

PR pipelinerun01a Running Actions

Overview YAML Logs

[Download](#) | [Expand](#)

- ✓ code compile
- ✓ compile & test
- ✓ unit test
- ✓ security check
- 🔄 image build

```
image build

[Plack::Sandbox::_2foft_2fapp_2droot_2fsrc_2fbin_2fapp_2epsi:54] core @2018-08-23 18:28:53> looking
for get /health in extlib/lib/perl5/Dancer2/Core/App.pm l. 36
[Plack::Sandbox::_2foft_2fapp_2droot_2fsrc_2fbin_2fapp_2epsi:54] core @2018-08-23 18:28:53> Entering
hook core.error.init in (eval 306) l. 1
[Plack::Sandbox::_2foft_2fapp_2droot_2fsrc_2fbin_2fapp_2epsi:54] core @2018-08-23 18:28:53> Entering
hook core.error.before in (eval 306) l. 1
[Plack::Sandbox::_2foft_2fapp_2droot_2fsrc_2fbin_2fapp_2epsi:54] core @2018-08-23

[Plack::Sandbox::_2foft_2fapp_2droot_2fsrc_2fbin_2fapp_2epsi:54] core @2018-08-23 18:28:53> looking
for get /health in extlib/lib/perl5/Dancer2/Core/App.pm l. 36
[Plack::Sandbox::_2foft_2fapp_2droot_2fsrc_2fbin_2fapp_2epsi:54] core @2018-08-23 18:28:53> Entering
hook core.error.init in (eval 306) l. 1
[Plack::Sandbox::_2foft_2fapp_2droot_2fsrc_2fbin_2fapp_2epsi:54] core @2018-08-23 18:28:53> Entering
hook core.error.before in (eval 306) l. 1
[Plack::Sandbox::_2foft_2fapp_2droot_2fsrc_2fbin_2fapp_2epsi:54] core @2018-08-23
```

Red Hat OpenShift Container Platform

Project: pipelines-demo Application: all applications

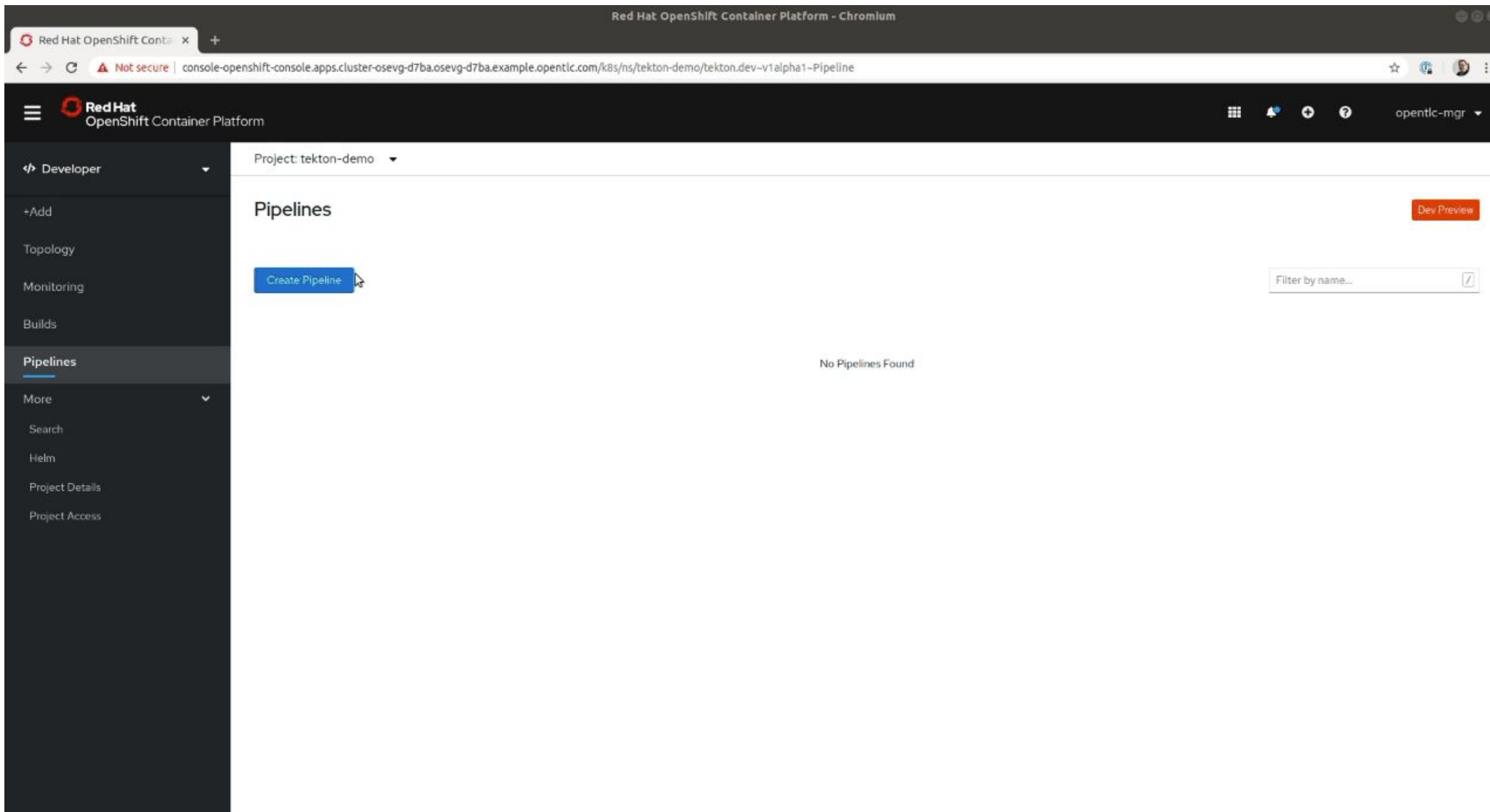
Add

No workloads found

To add content to your project, create an application, component or service using one of these options.

- From Git**
Import code from your git repository to be built and deployed
- Container Image**
Deploy an existing image from an image registry or image stream tag
- From Catalog**
Browse the catalog to discover, deploy and connect to services
- From Dockerfile**
Import your Dockerfile from your git repo to be built & deployed
- YAML**
Create resources from their YAML or JSON definitions
- Database**
Browse the catalog to discover database services to add to your application

Pipeline Builder to compose Pipelines from Task library



Manage Pipelines with Tekton CLI

```
ssadeghi@Siamaks-MacBook-Pro: ~ (zsh)
[~]$ # brew tap tektoncd/tools
[~]$ # brew install tektoncd/tools/tektoncd-cli
[~]$
[~]$
```

Tekton Pipelines VS Code Extension

The screenshot displays the VS Code interface with the Tekton Pipelines extension. The left sidebar shows a tree view of pipeline components: Pipelines, Tasks, ClusterTasks, PipelineResources, TriggerTemplates, TriggerBinding, EventListener, and Conditions. The main editor shows the YAML code for a pipeline named 'demo-pipeline'. The right sidebar shows a graphical flow diagram of the pipeline tasks.

```
pipeline << sample_test.yaml > {} spec > [] tasks
1  apiVersion: tekton.dev/v1alpha1
2  kind: Pipeline
3  metadata:
4    name: demo-pipeline
5  spec:
6    resources:
7      - name: source-repo
8        type: git
9      - name: web-image
10       type: image
11      - name: app-image
12        type: image
13  tasks:
14    - name: skaffold-unit-tests
15      taskRef:
16        name: unit-tests
17      resources:
18        inputs:
19          - name: workspace
20            resource: source-repo
21    - name: build-skaffold-web
22      runAfter: [skaffold-unit-tests]
23      taskRef:
24        name: build-push
25      params:
26        - name: pathToDockerFile
27          value: Dockerfile
28        - name: pathToContext
29          value: /workspace/workspace/examples/microservice
30      resources:
31        inputs:
32          - name: workspace
33            resource: source-repo
34        outputs:
35          - name: builtImage
36            resource: web-image
37    - name: build-skaffold-app
38      runAfter: [skaffold-unit-tests]
39      taskRef:
40        name: build-push
41      params:
42        - name: pathToDockerFile
43          value: Dockerfile
44        - name: pathToContext
45          value: /workspace/workspace/examples/microservice
46      resources:
```

The graphical flow diagram on the right shows the pipeline tasks as follows:

```
graph TD
  A[skaffold-unit-tests] --> B[build-skaffold-web]
  A --> C[build-skaffold-app]
  B --> D[deploy-web]
  C --> E[deploy-app]
```

GitOps

What is GitOps?

Git as the source of truth for the state of the entire system. Changes in Git drive changes in the system

Repeatable

Predictable

Auditable

Accessible

GitOps Workflow

desired
state



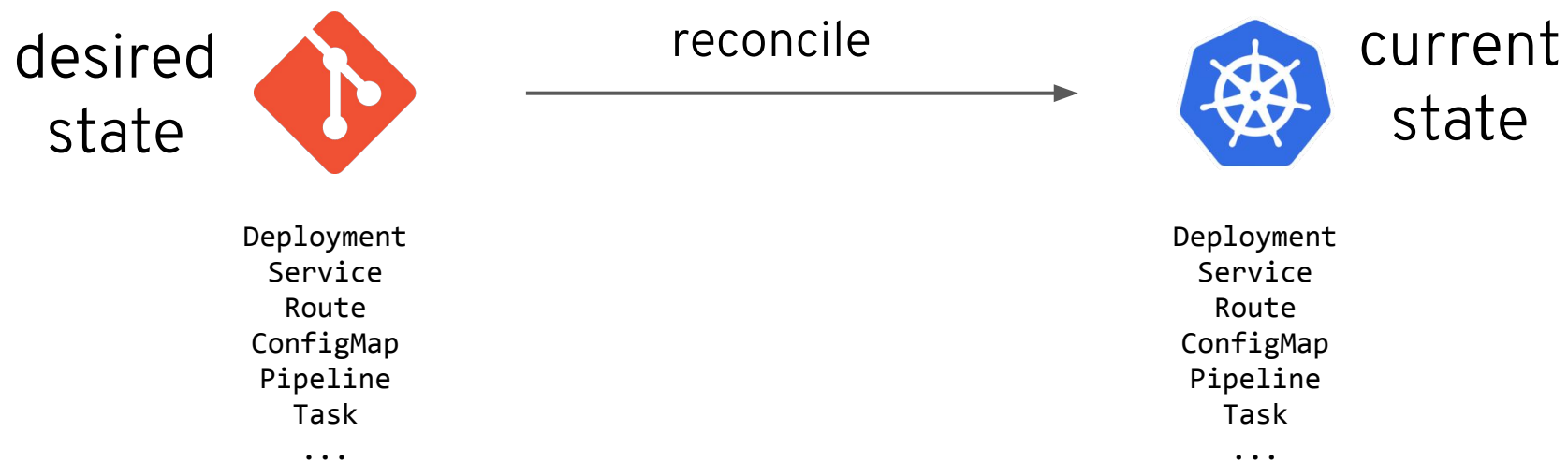
Deployment
Service
Route
ConfigMap
Pipeline
Task
...



current
state

Deployment
Service
Route
ConfigMap
Pipeline
Task
...

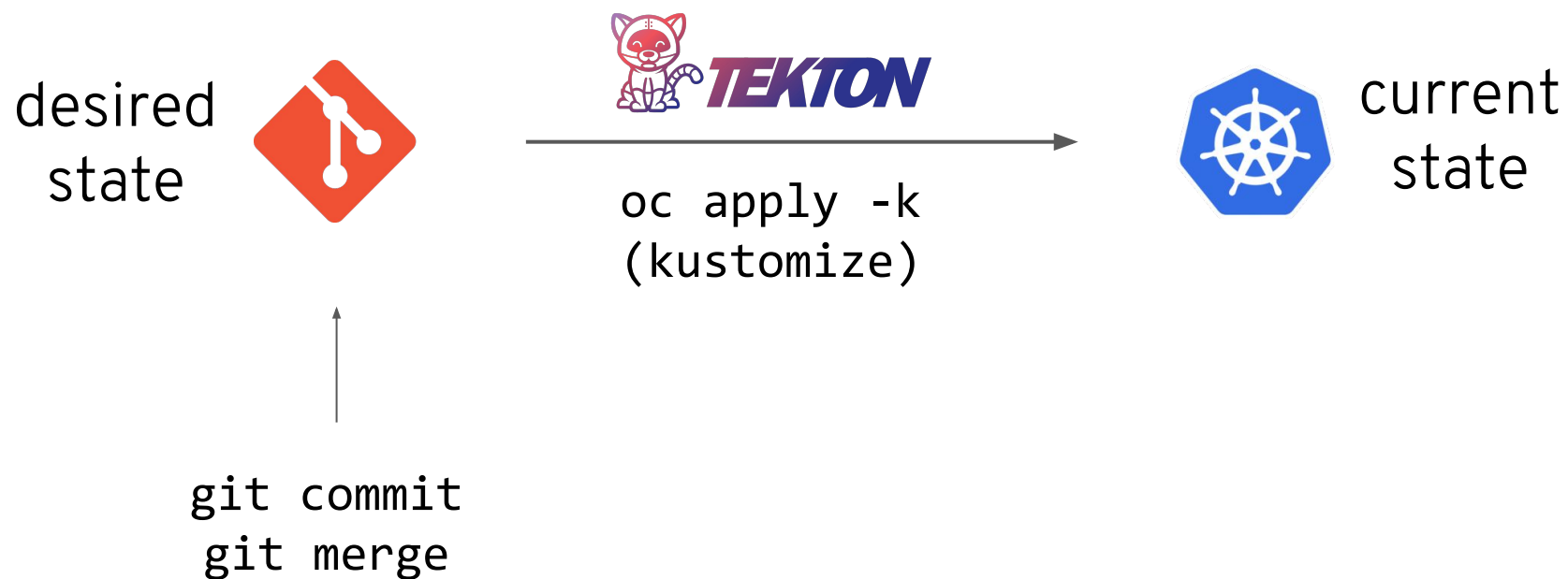
GitOps Workflow



GitOps Workflow



GitOps Workflow



GitOps Tooling



Argo CD

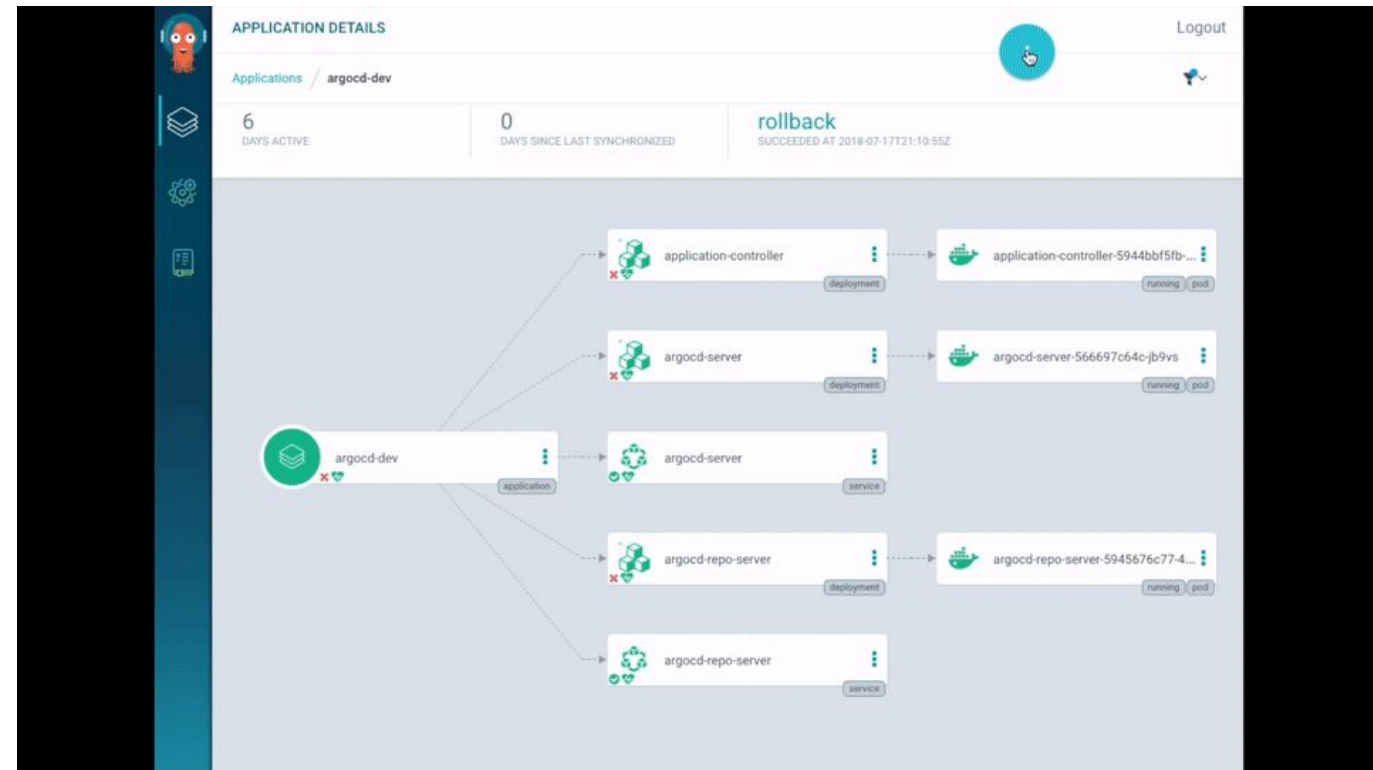


Kustomize

Argo CD - What is It?

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.

- Easily deploy applications in a declarative way
- Synchronizes cluster state with git repos
- Works with a variety of Kubernetes deployment tools including:
 - Helm
 - Kustomize
 - Ksonnet/Jsonnet
 - Directories of yaml
- **It is not a CI tool**



Argo CD - Synchronizing



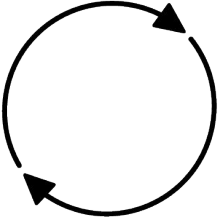
Change in git



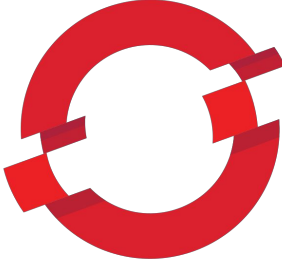
Pushed to Argo CD



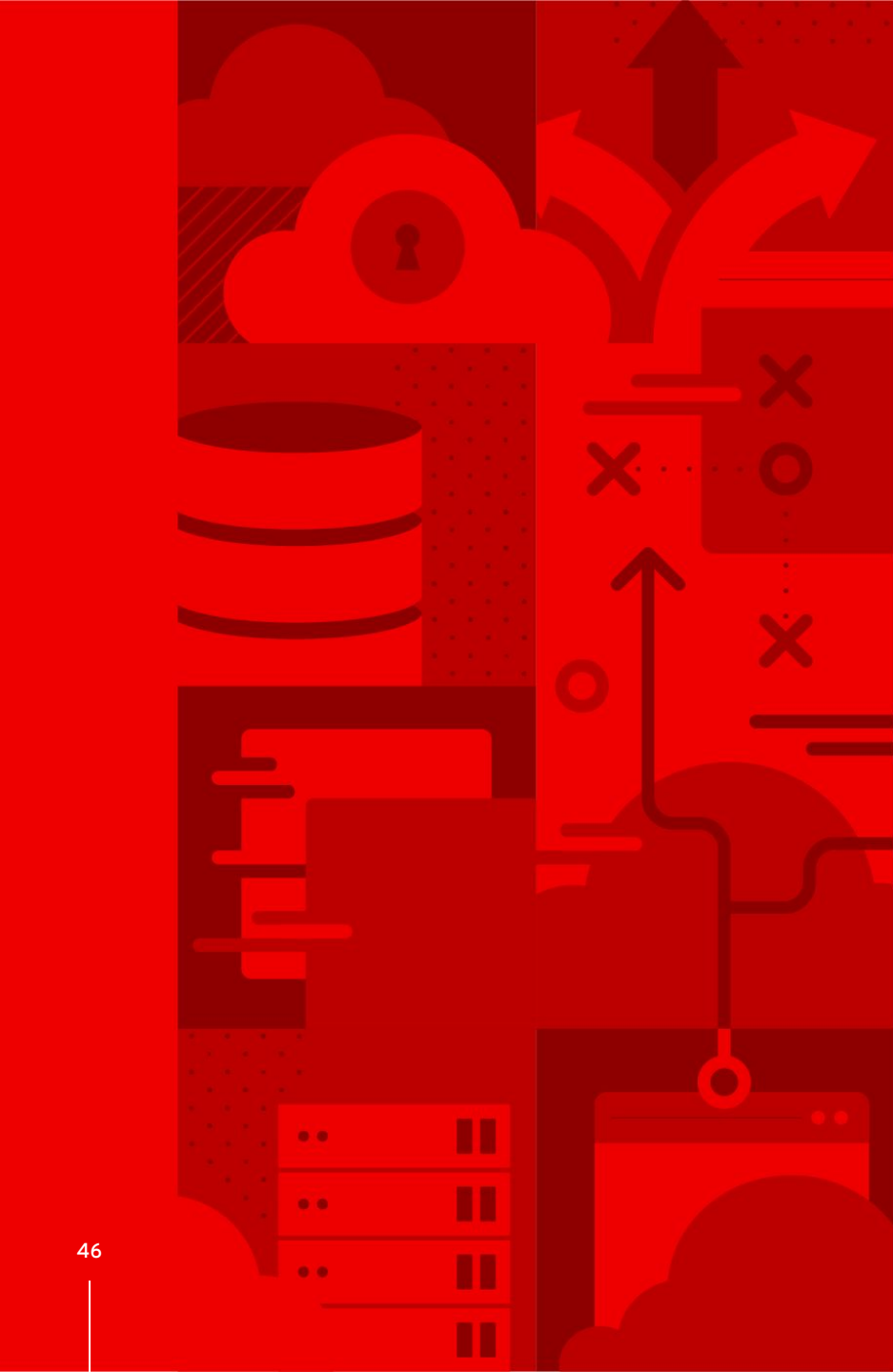
Check Sync Status



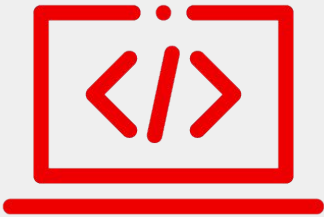
Synchronize



OPENSHIFT



Live Demo



Lessons Learned and Roadmap



Learnings

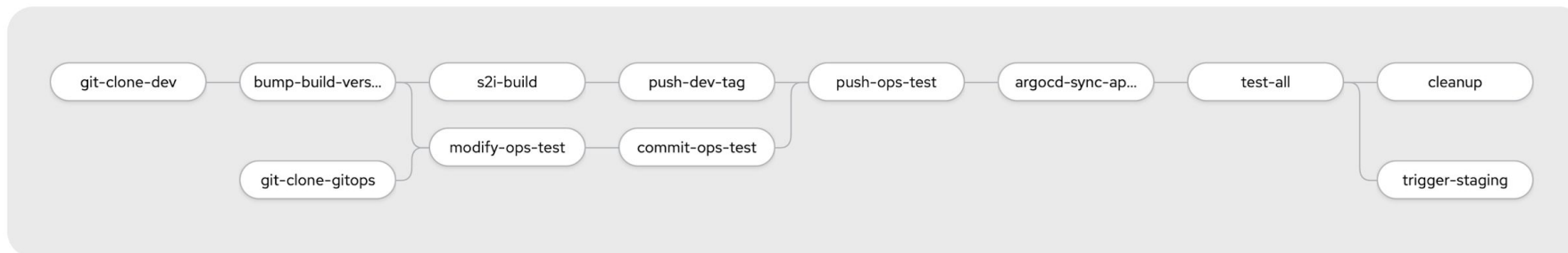
- We used Tekton in some internal projects
- Feels native because it is!

[Pipelines](#) > Pipeline Details

PL build-and-test

[Details](#) [YAML](#) [Pipeline Runs](#) [Parameters](#) [Resources](#)

Pipeline Details



Learnings

- Feels uncommon at first
- A lot of writing bash scripts and building container images
- Debug? Metrics? Human Input? SCM Connect?
- Sometimes not that easy to write tasks generic

Learnings

- Moderate UI experience
 - VSCode plugin, good but...
 - Tekton Dashboard, good but...
 - OpenShift Dev Console (Pipeline Builder), good but...

Learnings

- OLM update broke Tekton Operator
- Handling Task Results
- Some CRDs went to beta, Pipeline Resources -> Future?
- A lot of resources left after a build (Pods, PVCs etc.). Where is my build history?
- Triggering Subpipelines -> Results?

Why it's great?

- Cloud native! **Declarative config, easy to share, easy to understand**
- Environment promotion through **GitOps**
- Member of cd.foundation and part of Jenkins X, Kabanero, Relay...
- **Task Catalog** and new: Tekton Hub
- Foundational platform for Continuous Delivery
- Focus on **core API** -> enable 3rd party tooling
- **Reusable!! Serverless!**
- Will be a **supported "product" within OpenShift!!**

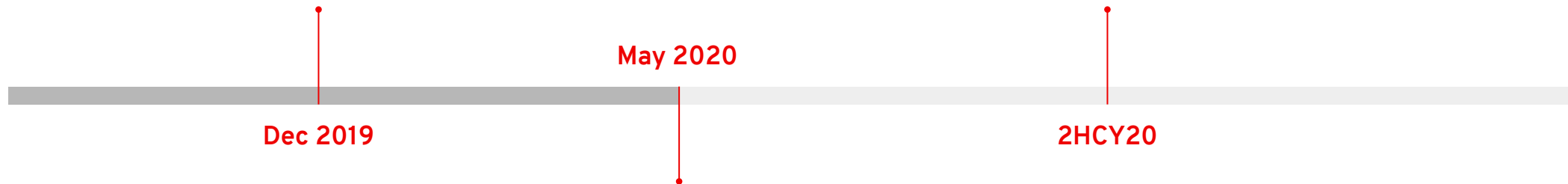
OpenShift Pipelines Roadmap

Dev Preview 4

- Webhooks
- Default tasks
- RBAC UX improvements (cont.)
- Tekton CLI (more commands)
- VSCode Plugin (cont.)
- Console Dev (view, graph, logs)

GA

- Disconnected install
- Proxy support
- Image stream support
- Support for finally clauses
- Enhanced pipeline resources
- Console - catalog integration



Dec 2019

May 2020

2HCY20

Tech Preview

- Webhook event filtering
- More tasks in the catalog
- Console - generate pipelines
- Console - task snippets
- Console - pipeline builder
- Workspace and sharing artifacts
- Tekton CLI (more commands)
- VSCode Plugin (cont.)

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat