



OPENSIFT CI/CD

NEUEDA.COM

Objectives

- OpenShift CI/CD
- OpenShift Builds and OpenShift GitOps
- OpenShift Pipelines and Tekton
- The OpenShift Pipeline Operator
- Pipeline Triggers with Webhooks

CI/CD in OpenShift

- OpenShift provides the following CI/CD solutions
 - OpenShift Builds
 - OpenShift Pipelines
 - OpenShift GitOps
 - External CI/CD, e.g. Jenkins, GitHub Actions
- We will briefly look at OpenShift Builds, GitHub Actions and GitOps, but will focus on OpenShift Pipelines.

BUILDS, ACTIONS AND GITOPS

OpenShift Builds

- OpenShift Builds facilitate creating cloud-native apps using a declarative build process.
- The build process can be defined in a YAML file that you use to create a BuildConfig object.
 - The definition includes attributes like build triggers, input parameters and source code.
- When deployed the BuildConfig object typically builds a runnable image and pushes it to a container image registry.

OpenShift Build Strategies

- OpenShift 4.10 Builds supports
 - **Docker build:** Buildah is used to build a container image from a Dockerfile.
 - **Source-to-Image (s2i):** This produces a ready-to-run image by injecting application source into a container image and assembling a new image.
 - **Custom build:** This gives you the flexibility to customize your own build process.

OpenShift Build Triggers

- OpenShift BuildConfigs are provisioned to accept Webhooks out of the box.
- The Payload URL can be found at the bottom of the BuildConfig details:

The image shows two screenshots of the OpenShift console interface, connected by a large orange arrow pointing from left to right. The left screenshot shows the 'BuildConfigs' list for the 'ose10-nationalparks' project. The 'Builds' menu item in the left sidebar is circled in red, and the 'nationalparks' BuildConfig entry in the table is also circled in red. The right screenshot shows the details for the 'nationalparks' BuildConfig. The 'Webhooks' tab is circled in red. Below the tab, there is a table with two entries: 'Generic' and 'GitHub'. Each entry has a 'Type', a 'Webhook URL', and a 'Secret'.

Type	Webhook URL	Secret
Generic	https://api.ose.openshift.com/6443/apis/build.openshift.io/v1/namespaces/ose10-nationalparks/buildconfigs/nationalparks/webhooks/secret/generic	nationalparks-generic-webhook-secret Copy URL with Secret
GitHub	https://api.ose.openshift.com/6443/apis/build.openshift.io/v1/namespaces/ose10-nationalparks/buildconfigs/nationalparks/webhooks/secret/github	nationalparks-github-webhook-secret Copy URL with Secret

OpenShift GitOps

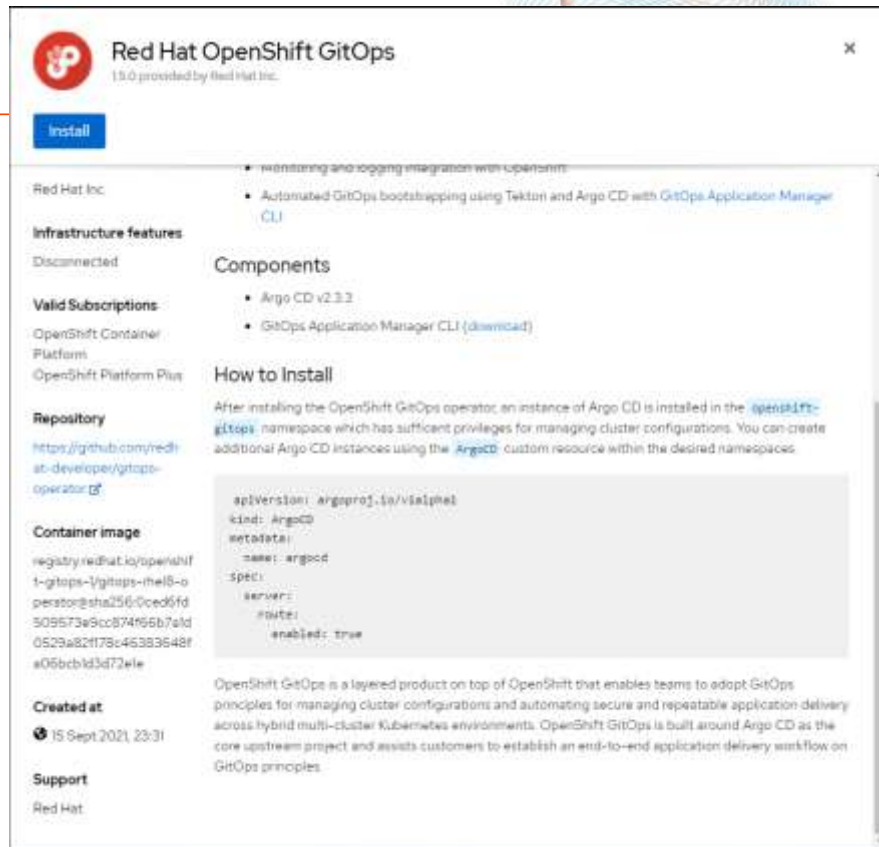
- GitOps is a declarative way to implement continuous deployment for cloud native applications.
- The Git repository becomes the sole source of truth
 - The workflow pushes changes to the repository through testing, staging and production.
 - You only need to update the repository, GitOps does everything else.

OpenShift GitOps Repositories

- OpenShift GitOps always has at least two repositories
 - Application repository with the source code
 - Environment configuration repository defining the desired state of the application
- Together these repositories contain a declarative description of the infrastructure you need in your environment.
- OpenShift uses [Argo CD](#) to maintain cluster resources.
 - Argo CD is an open-source declarative tool for CI/CD

Enabling OpenShift GitOps

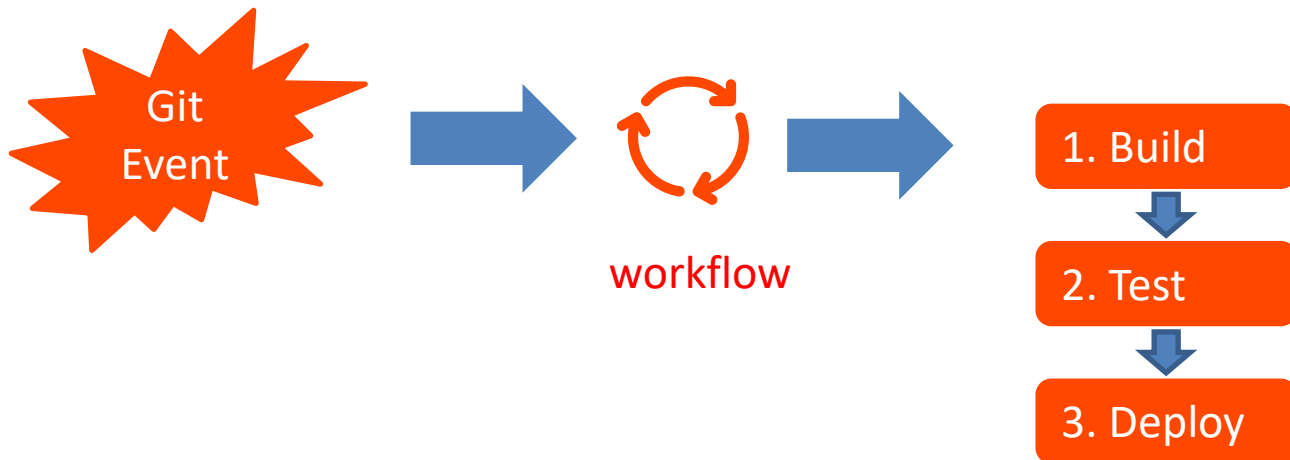
- OpenShift GitOps is enabled through an Operator on the OperatorHub.
- Once installed you can log into the Argo CD UI and create an Argo CD GitOps application.



<https://github.com/siamaksade/openshift-gitops-getting-started>

External CI/CD: GitHub Actions, Jenkins etc

- Event-driven automation tasks hooked into a git repo
- Triggered by events in the repo
- All happens *outside* of OpenShift - drives OpenShift deployment



OPENSIFT PIPELINES

OpenShift Pipelines

- OpenShift integrates a cloud native solution for CI/CD based on the open source Tekton project.
- Tekton
 - Runs on the same Kubernetes backbone as OpenShift
 - Has a set of custom k8s resources that give you the building blocks for your CI/CD pipelines.

Tekton

- Tekton building blocks are called **steps**.
- You collect Tekton steps into **tasks**, such as test, build and deploy.
- Tasks are organised into **pipelines**, which are tasks ordered either sequentially or concurrently.



Tekton

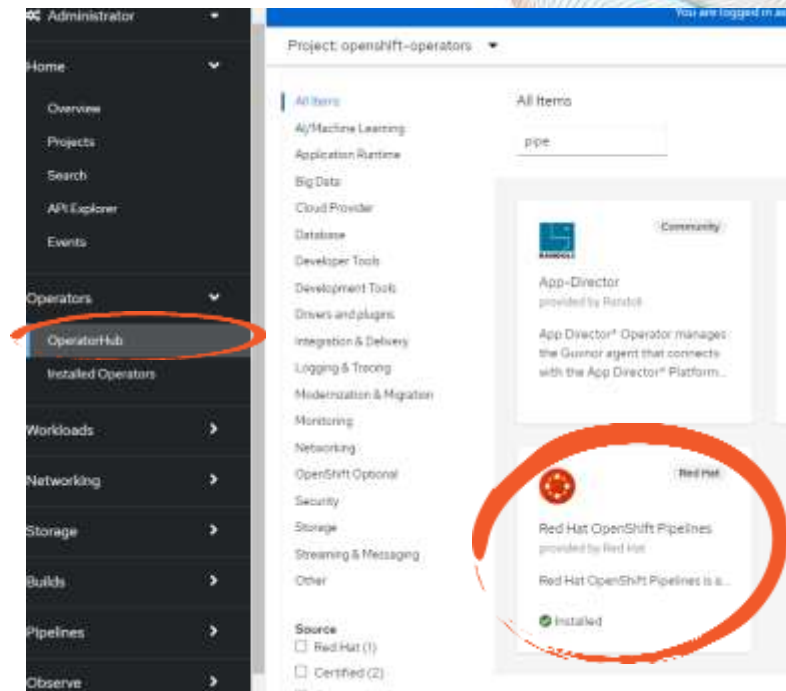
- Pipelines are triggered through a **trigger template**, that is designed to respond to various events, such as a git push.
- The events are registered by an **event listener**, which interfaces with the trigger template.
- Each of the concepts (step, task, pipeline, trigger template, event listener) are just files (yaml format)
 - They can be put into source control
 - They can be easily reused.

Tekton Workspaces


- Tekton collects its CI/CD data into workspaces.
- These use a native k8s storage concept called a persistent volume claim.
 - Usually defined in the trigger templates, along with secrets and authorization.
- OpenShift Pipelines *is* Tekton
 - The two terms are used interchangeably in OpenShift documentation.

OpenShift Pipelines Operator

- OpenShift OperatorHub is a catalogue of available Operators
 - This contains OpenShift extensions and tools for a variety of common activities
- The **Red Hat OpenShift Pipelines Operator** has been installed from here

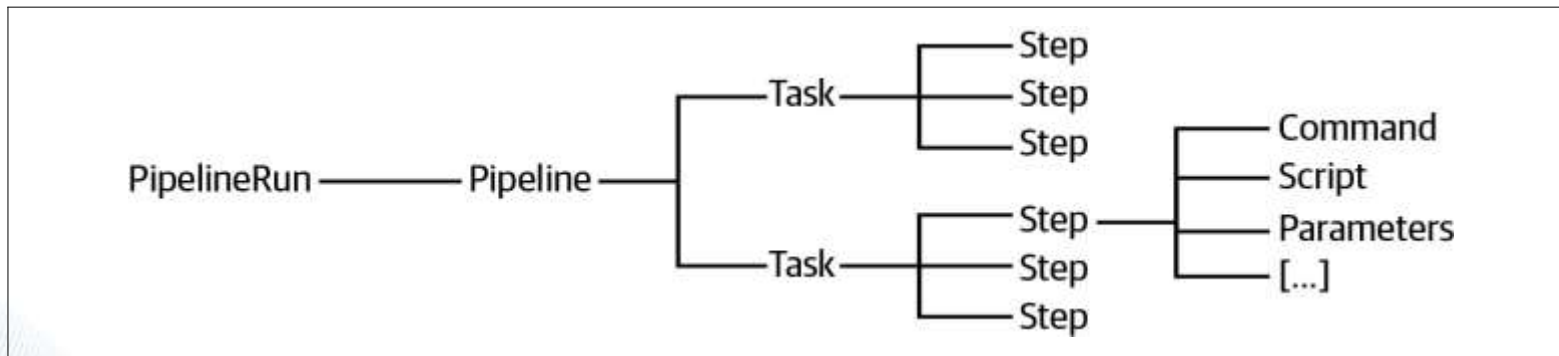


Tekton in OpenShift

- In the Developer perspective of the Web Console:
 - You can create pipeline tasks
 - You can assemble tasks into pipelines
 - You can run and monitor pipelines.
- You can do the same in the command line using a tool called `tkn`
 - Download it from the  menu on the Console

Tekton in OpenShift

- Tekton uses a TriggerTemplate object coupled with an EventListener object to trigger automatic runs. OpenShift uses the same mechanism.
- Remember a step is a sequence of commands to achieve a specific goal, like building a container image.



Pipeline Steps: Command

- A pipeline Step consists of either a command or a script along with a set of parameters.
- For example, consider the command:

- name: generate	Name of the command
------------------	---------------------

command:

- s2i	Utility to run (Source-to-Image here)
-------	---------------------------------------

- build	Utility subcommand
---------	--------------------

- \$(params.PATH_CONTEXT)	Set the PATH_CONTEXT parameter to the given image
---------------------------	---

- registry.access.redhat.com/redhat-openjdk-18/openjdk18-openshift	
--	--

- '--image-scripts-url'	Argument for the s2i build subcommand
-------------------------	---------------------------------------

- 'image:///usr/local/s2i'	
----------------------------	--

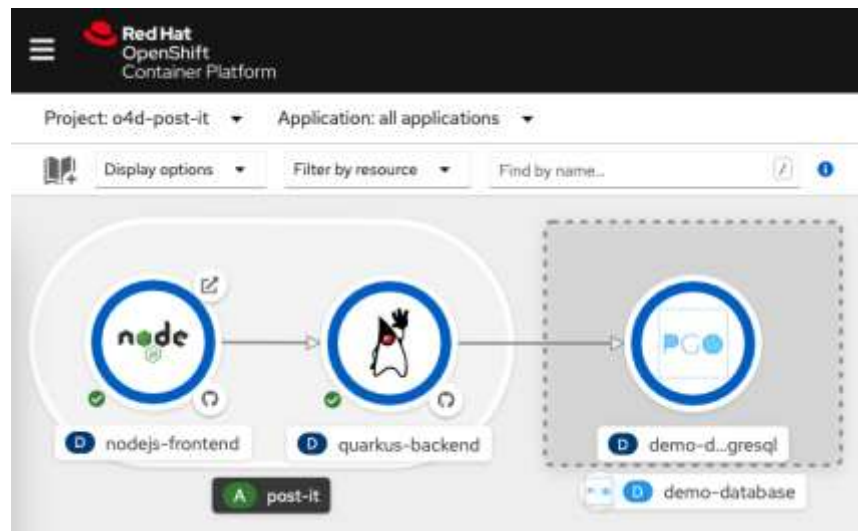
Pipeline Steps: Script

- Suppose several operations must be run to achieve a specific Step, then a Script is a better option.
- A script puts an executable script inline.
- For example in this script the python3 interpreter is invoked to run the subsequent commands.

```
- name: lint-markdown
  script: |-
    #!/usr/bin/env python3
    ...
```

Demo Example: A Cloud-Ready Notes Application

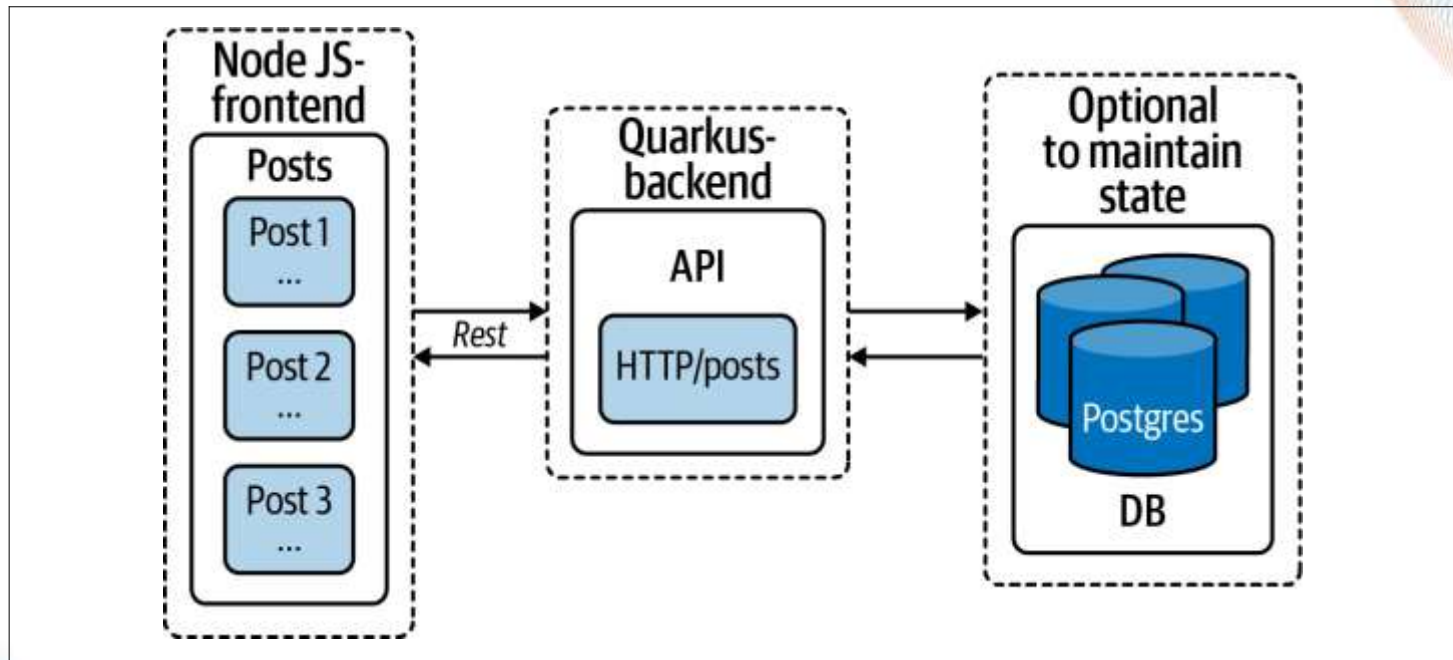
- This is a more sophisticated application that should eventually include a database (not demoed).



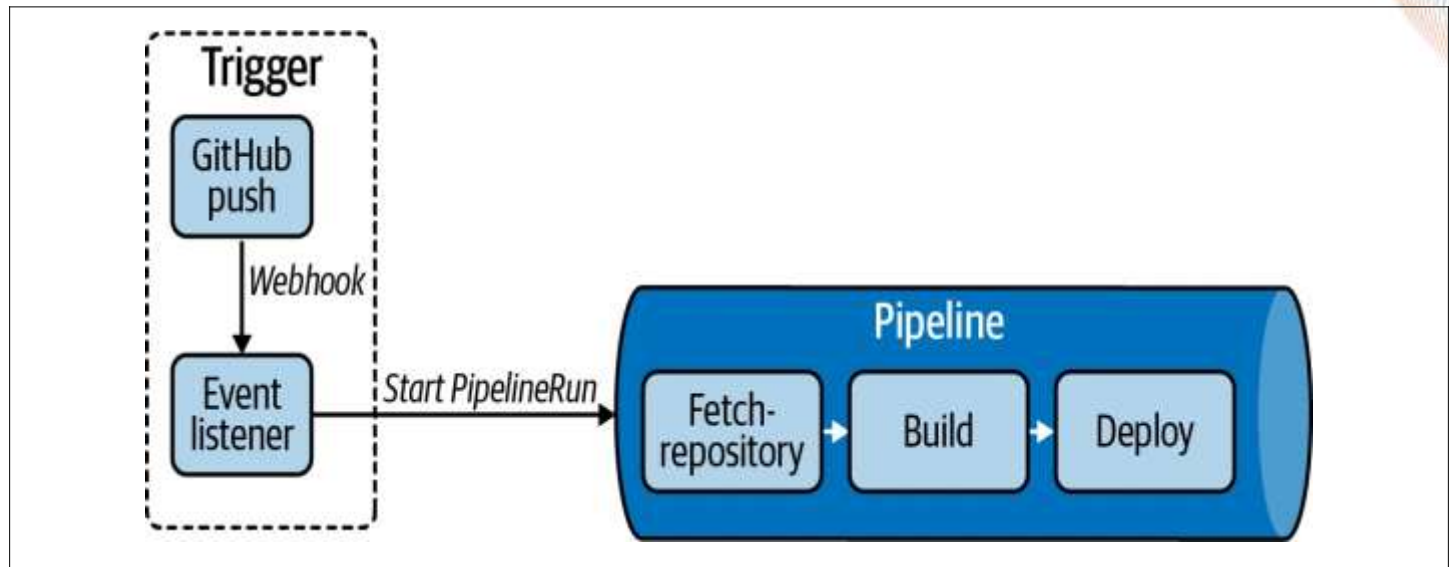
Demo Example : A Cloud-Ready Notes Application

- The example is a simple note board
 - Each note is a title and content
- A database can be connected, which will allow you to maintain and manage prior posts.
- The frontend is written in Node.js and uses React.
 - We won't be editing this, we can deploy straight from GitHub.
- The backend is written using Quarkus (a k8s-native Java stack)

Application Topology



Pipeline Configuration



Demo Application

- We'll work this through together on the OpenShift cluster!

Demo Summary

- In this demo we:
 - Used the Pipelines operator to create a build Pipeline for an application.
 - Configured a Pipeline Trigger to respond to a GitHub push event.
 - Demonstrated an automatic pipeline run in response to the webhook event.

Lab Exercise 2

- Do your own Pipeline deployment!

Summary

- OpenShift CI/CD
- OpenShift Builds and OpenShift GitOps
- OpenShift Pipelines and Tekton
- The OpenShift Pipeline Operator
- Pipeline Triggers with Webhooks

Questions and Comments?

