

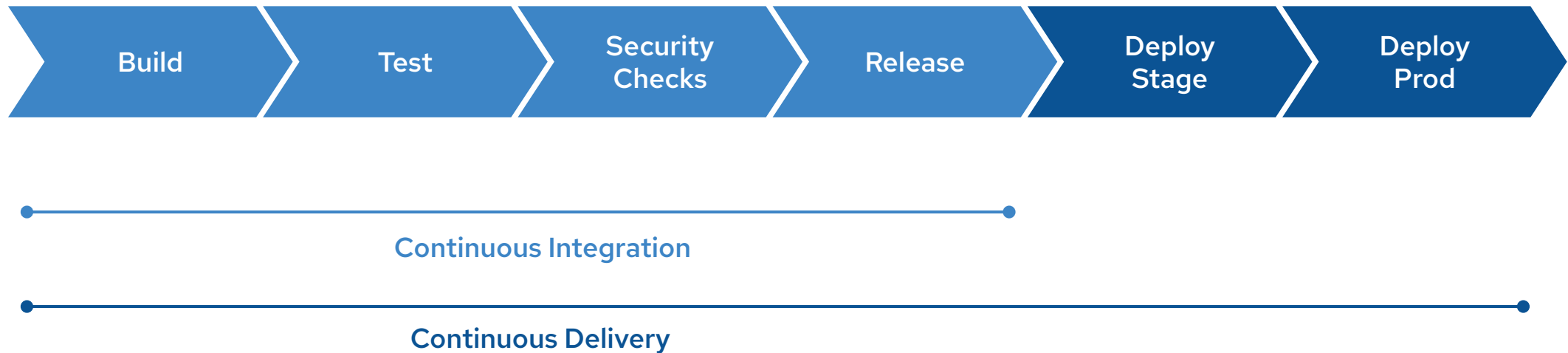
# Continuous Delivery and GitOps on OpenShift

Wanja Pernath

DevOps is the key to meet the  
insatiable demand for delivering quality  
applications rapidly

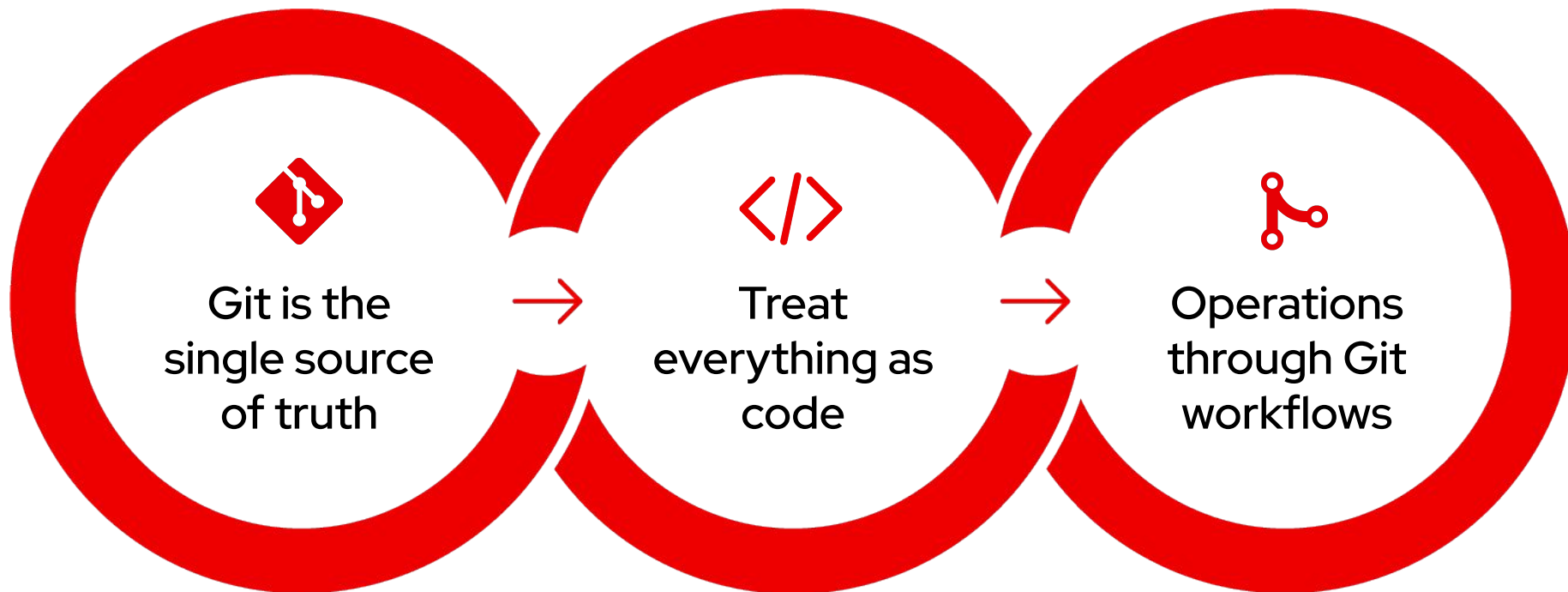
# Continuous Integration(CI) & Continuous Delivery (CD)

A key DevOps principle for automation, consistency and reliability



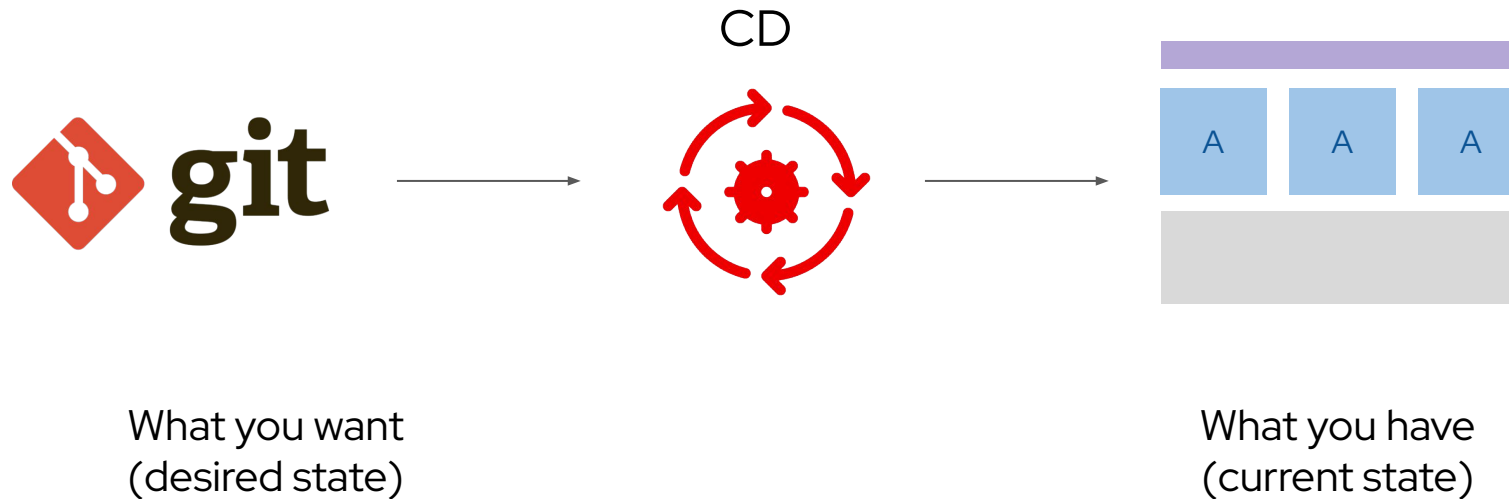
# What is GitOps?

An developer-centric approach to Continuous Delivery and infrastructure operation



# GitOps Workflow

a declarative approach to application delivery



# Why GitOps?

## Standard Workflow

Familiar tools and Git workflows from application development teams

## Enhanced Security

Review changes beforehand, detect configuration drifts, and take action

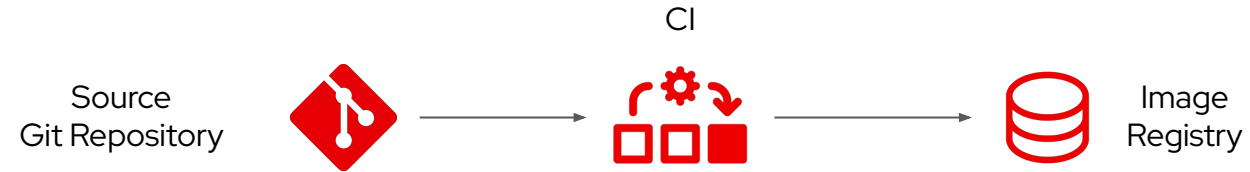
## Visibility and Audit

Capturing and tracing any change to clusters through Git history

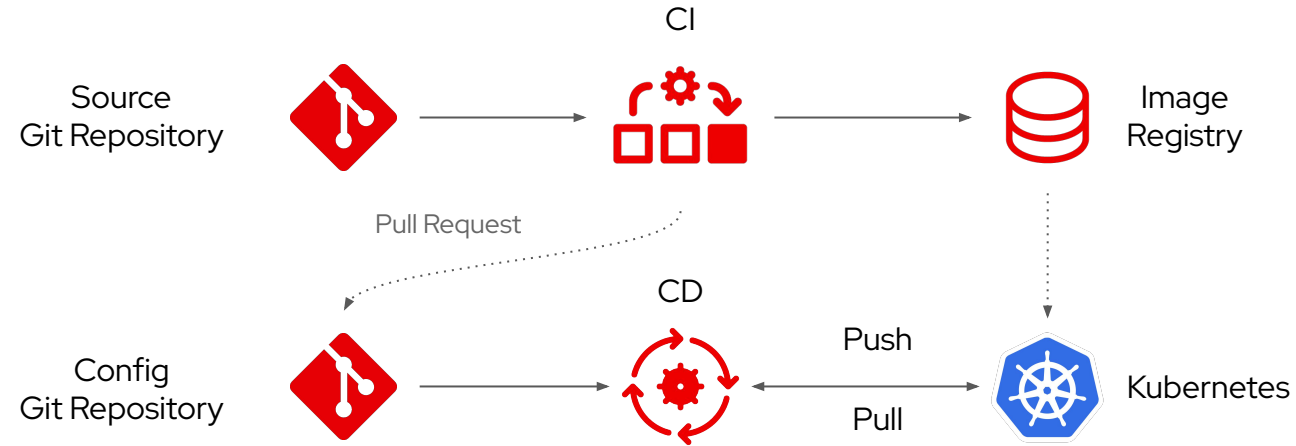
## Multi-cluster consistency

Reliably and consistently configure multiple Kubernetes clusters and deployment

# The GitOps Application Delivery Model

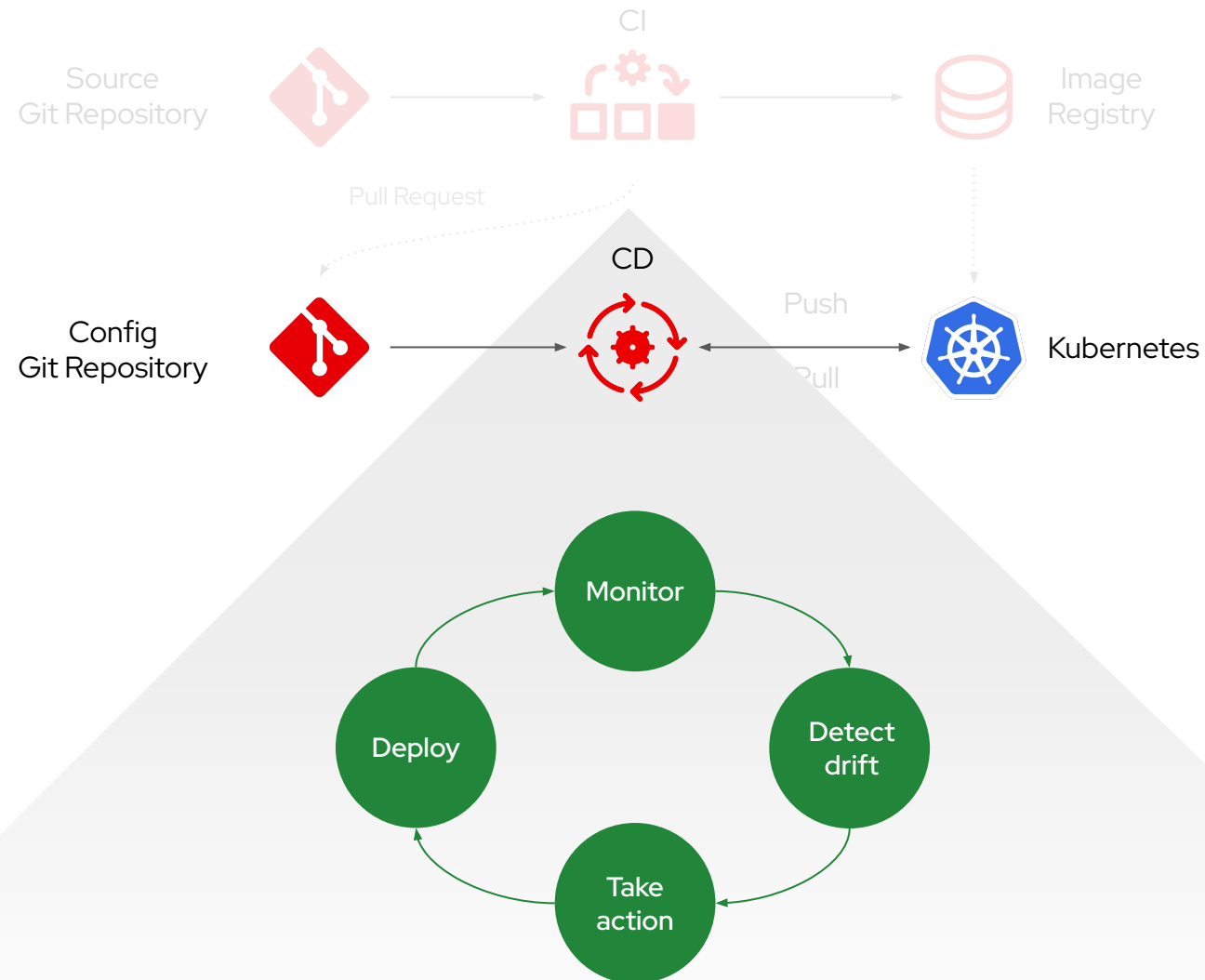


# The GitOps Application Delivery Model





# The GitOps Application Delivery Model



# A Comprehensive DevOps Platform for Hybrid Cloud

Automate building  
container images using  
Kubernetes tools



**OpenShift  
Builds**

Kubernetes-native  
on-demand delivery  
pipelines



**OpenShift  
Pipelines**

Declarative GitOps for  
multi-cluster continuous  
delivery



**OpenShift  
GitOps**

**OpenShift**



Azure DevOps



Jenkins

# Continuous Integration & Continuous Delivery



## OpenShift Build

Automate building container images using Kubernetes tools

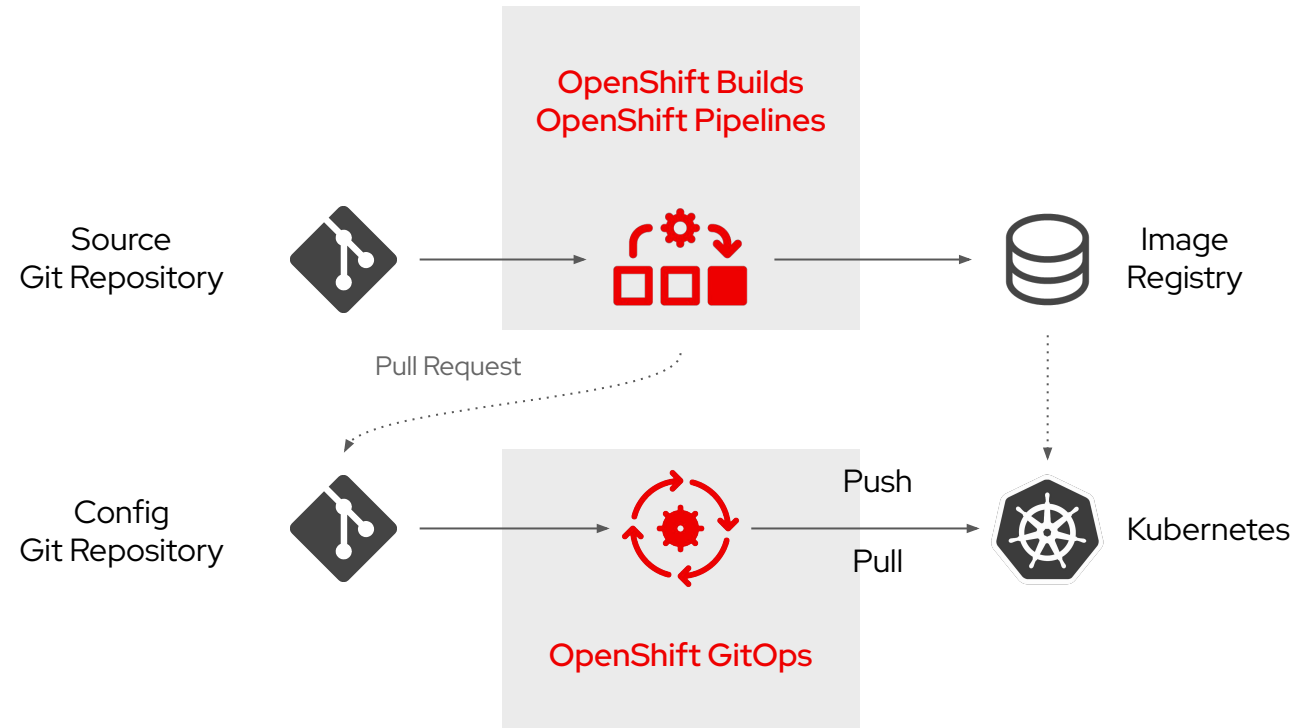
## OpenShift Pipelines

Kubernetes-native on-demand delivery pipelines

## OpenShift GitOps

Declarative GitOps for multi-cluster continuous delivery

# The GitOps Application Delivery Model on OpenShift



# OpenShift Builds

Automate building container images  
using Kubernetes tools

# OpenShift Builds



## Kubernetes-native image build

A Kubernetes-native way to building container images on OpenShift which is portable across Kubernetes distros



## Supports multiple build strategies

Choose the build strategy that fits best your applications and skills: source-to-image, Dockerfile, and Cloud-Native Buildpacks

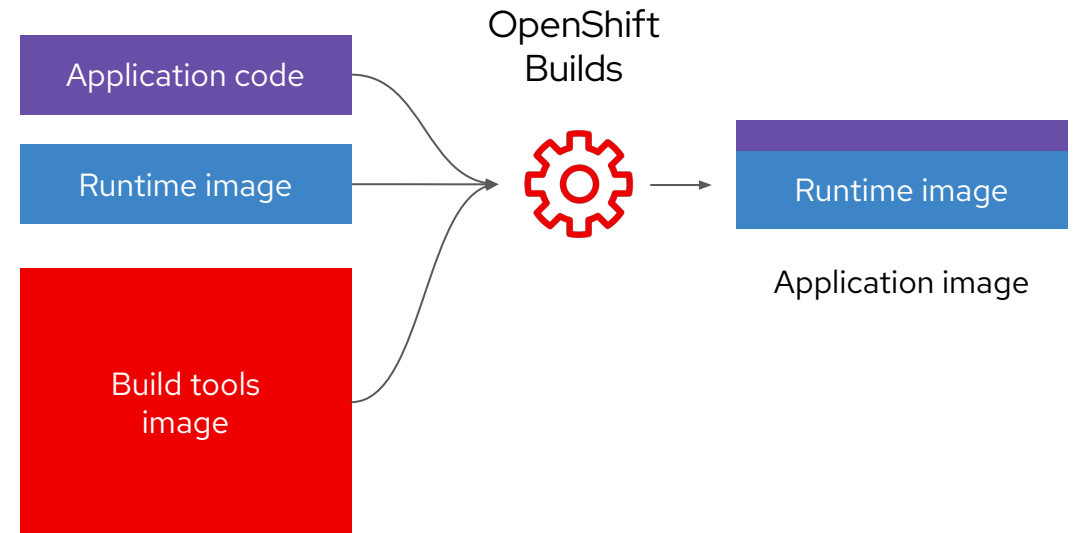


## Extend with additional build strategies

Extend to use community Kubernetes builds strategies or your own custom builds

# OpenShift Builds

- Build images on OpenShift and Kubernetes
- Use Kubernetes builds tools
  - Source-to-Image
  - Buildpacks
  - Buildah
  - Kaniko
  - ...more
- Create lean application images
- Extend with your own build tools
- Based on Shipwright open-source project



# OpenShift Builds

## **BuildStrategy**

How to build images e.g. S2I, Buildpacks, etc

## **Build**

What to build

## **BuildRun**

Build execution details



# OpenShift Builds

## Cloud-Native Buildpacks

```
kind: Build
metadata:
  name: myapp-buildpack
spec:
  source:
    url: https://github.com/myorg/myapp
  strategy:
    name: buildpacks-v3
  builder:
    image: paketobuildpacks/builder:full
  output:
    image: quay.io/myorg/myapp:v1
```

## Source-to-Image (S2I)

```
kind: Build
metadata:
  name: myapp-s2i
spec:
  source:
    url: https://github.com/myorg/myapp
  strategy:
    name: source-to-image
  builder:
    image: registry.redhat.io/openjdk/openjdk-11-rhel8
  output:
    image: quay.io/myorg/myapp:v1
  runtime:
    image: docker.io/openjdk:11-jre-slim
```

# OpenShift Builds v2 Roadmap

## OpenShift Builds v2 Dev Preview

- Build from application binary
- Kaniko build strategy

Q4 CY20

1H CY21

## OpenShift Builds v2 Tech Preview

- Builds in air-gapped clusters
- Proxy support
- Shipwright CLI

## OpenShift Builds v2 GA

- Supported S2I, Buildah and Buildpacks strategies
- Trigger builds from Git
- Unprivileged builds
- Volume support
- Integrated in Dev Console

2H CY21

# OpenShift Pipelines

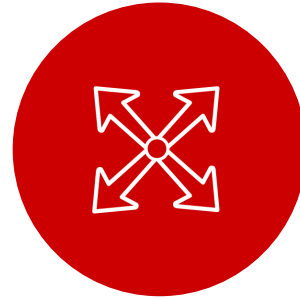
Kubernetes-native on-demand delivery  
pipelines

# 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 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



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



## Built for Kubernetes

Cloud-native pipelines taking advantage of Kubernetes execution and , operational model and concepts



## Scale on-demand

Pipelines run and scale on-demand in isolated containers, with repeatable and predictable outcomes



## Secure pipeline execution

Kubernetes RBAC and security model ensures security consistently across pipelines and workloads

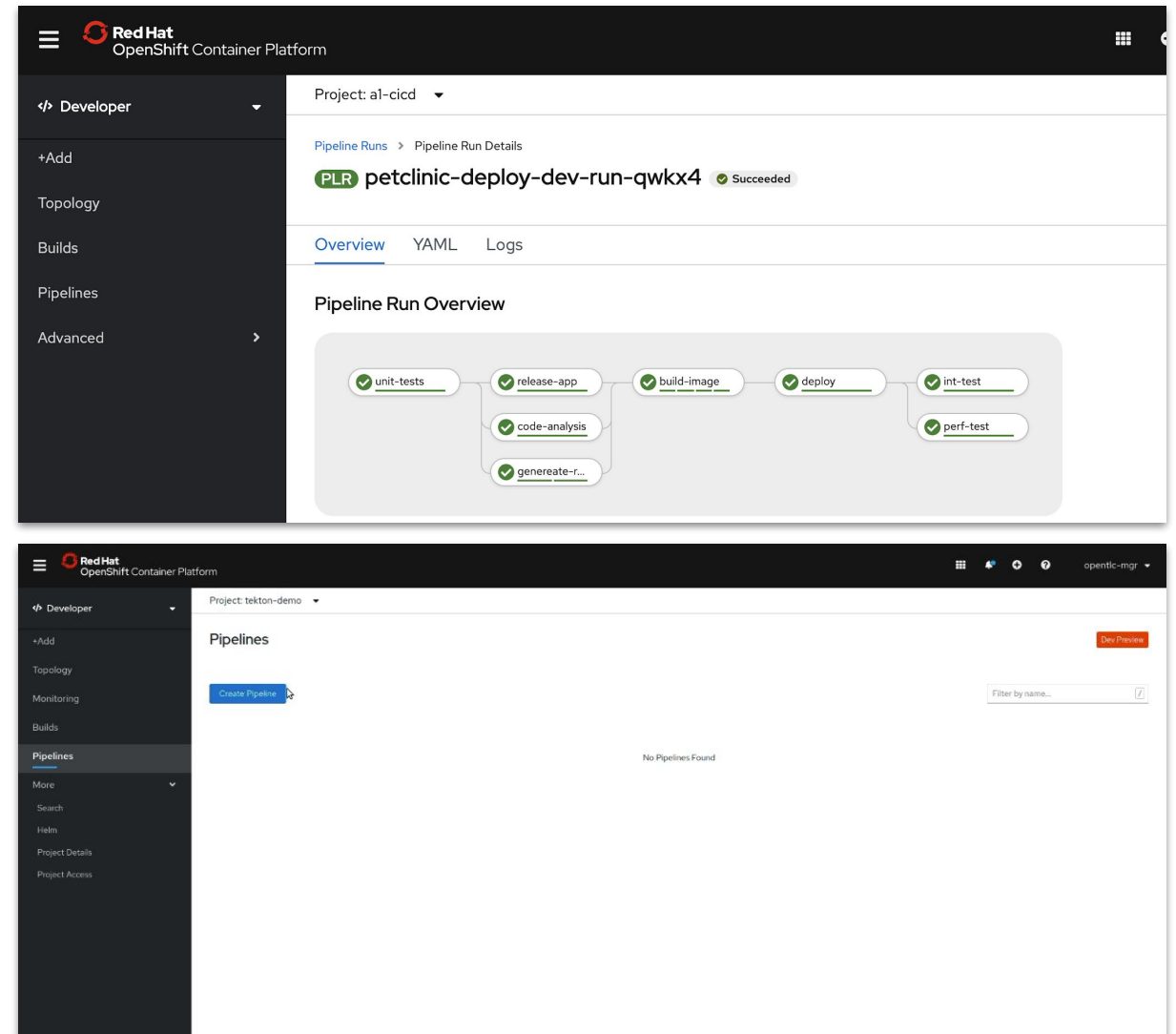


## Flexible and powerful

Granular control over pipeline execution details on Kubernetes, to support your exact requirements

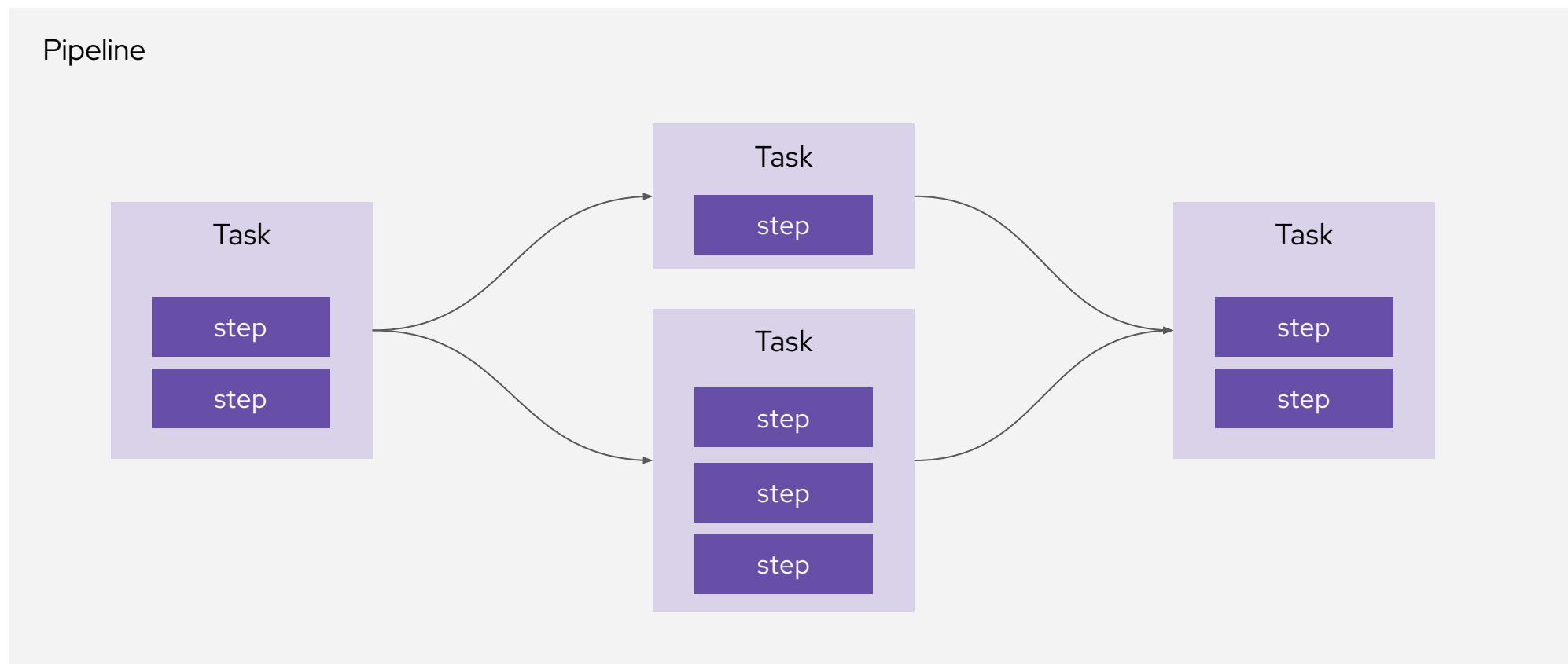
# OpenShift Pipelines

- Based on Tekton Pipelines
- Kubernetes-native declarative CI/CD
- Pipelines run on-demand in isolated containers
- No central server to maintain! No plugin conflicts!
- Task library and integration with Tekton Hub
- Secure pipelines aligned with Kubernetes RBAC
- Visual and IDE-based pipeline authoring
- Pipeline templates when importing apps
- Automated install and upgrades via OperatorHub
- CLI, Web, VS Code and IntelliJ plugins





# Tekton Concepts



# Tekton Concepts: step

- 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
    ...
```

# Tekton Concepts: Task

- Performs a specific task
- List of steps
- Steps run sequentially
- Reusable

```
kind: Task
metadata:
  name: buildah
spec:
  params:
    - name: IMAGE
  steps:
    - name: build
      image: quay.io/buildah/stable:latest
      command: ["buildah"]
      args: ["bud", ".", "-t", "${params.IMAGE}"]
    - name: push
      image: quay.io/buildah/stable:latest
      script: |
        buildah push $(params.IMAGE) docker://$(params.IMAGE)
```

# Tekton Hub

## Search, discover and install Tekton Tasks

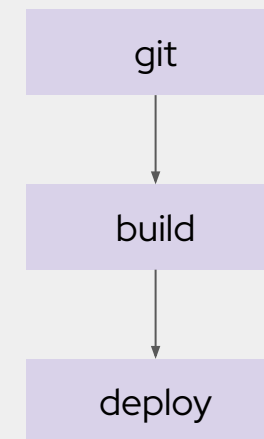
The screenshot displays the Tekton Hub (BETA) interface. At the top, there's a dark header with the Tekton Hub logo and a 'Login' link. Below the header, a large banner reads 'Welcome to Tekton Hub' and 'Discover, search and share reusable Tasks and Pipelines'. The main content area features a search bar and a 'Sort' dropdown set to 'Name'. On the left, there's a 'Refine By' sidebar with filters for 'Kind' (Task, Pipeline), 'Support Tier' (Official, Verified, Community), and 'Categories' (Build Tools, CLI, Cloud, Deploy, Image Build, Notification, Others, Test Framework). The main grid shows eight task cards, each with a Tekton logo, a user icon, a star rating, a title, a version number, a description, an update date, and tags. The tasks are: Ansible Runner (4.5 stars, v0.1, updated 3 weeks ago, tag: cli), ansible tower cli (2.0 stars, v0.1, updated 3 weeks ago, tags: ansible, cli), argocd (3.0 stars, v0.1, updated 3 weeks ago, tag: deploy), aws cli (5.0 stars, v0.1, updated 3 weeks ago, tag: cli), Amazon ECR Login (4.0 stars, v0.1, updated 3 weeks ago, tags: aws, ecr), azure cli (1.0 stars, v0.1, updated 4 months ago, tag: cli), bentoml (0.0 stars, v0.1, updated 3 weeks ago, tag: cli), and Python Black (0.0 stars, v0.1, updated 3 weeks ago, tags: formatter, python).

Task Name	Rating	Version	Updated	Tags
Ansible Runner	4.5	v0.1	3 weeks ago	cli
ansible tower cli	2.0	v0.1	3 weeks ago	ansible, cli
argocd	3.0	v0.1	3 weeks ago	deploy
aws cli	5.0	v0.1	3 weeks ago	cli
Amazon ECR Login	4.0	v0.1	3 weeks ago	aws, ecr
azure cli	1.0	v0.1	4 months ago	cli
bentoml	0.0	v0.1	3 weeks ago	cli
Python Black	0.0	v0.1	3 weeks ago	formatter, python

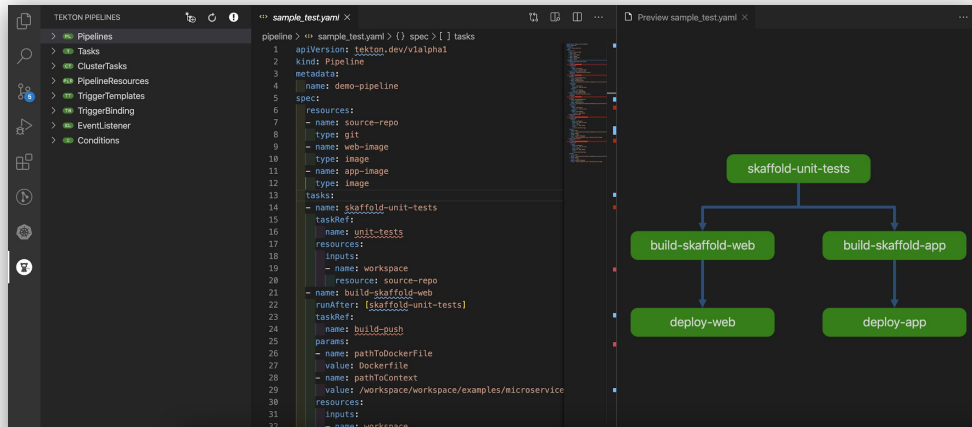
# Tekton Concepts: Pipeline

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

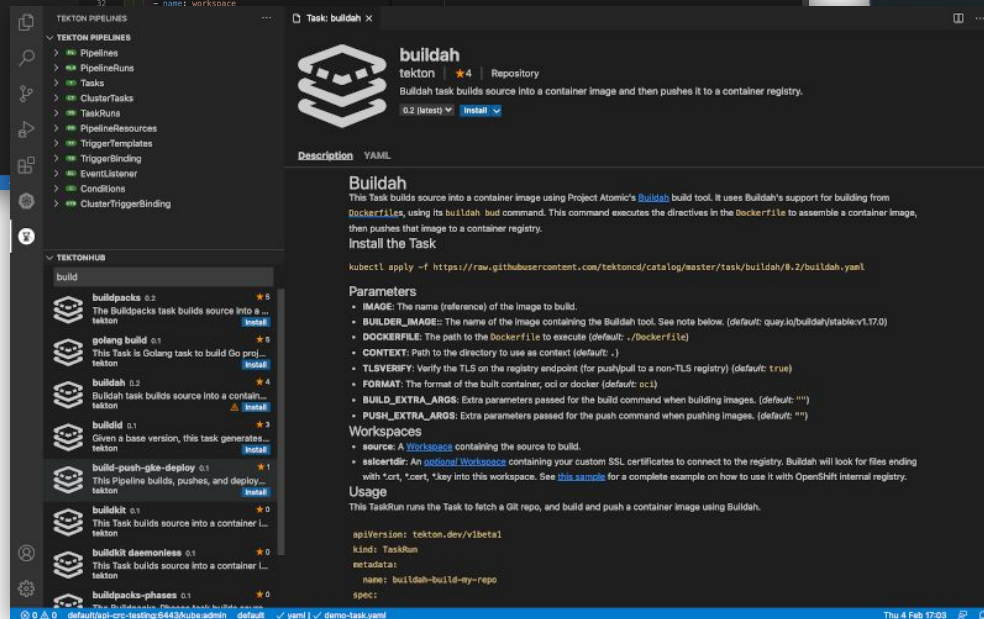
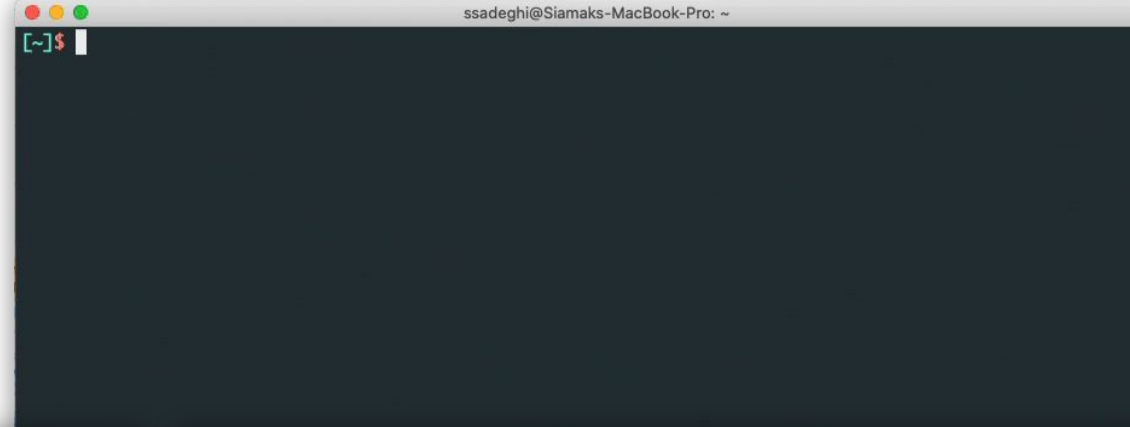
```
kind: Pipeline
metadata:
  name: deploy-dev
spec:
  params:
    - name: IMAGE_TAG
  tasks:
    - name: git
      taskRef:
        name: git-clone
        params: [...]
    - name: build
      taskRef:
        name: maven
        params: [...]
        runAfter: ["git"]
    - name: deploy
      taskRef: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



# Tekton CLI, Visual Studio Code, and IntelliJ



```
pipeline:
  apiVersion: tekton.dev/v1alpha1
  kind: Pipeline
  metadata:
    name: demo-pipeline
  spec:
    resources:
      - name: source-repo
        type: git
      - name: web-image
        type: image
      - name: app-image
        type: image
    tasks:
      - name: scaffold-unit-tests
        taskRef:
          name: scaffold-unit-tests
        resources:
          source-repo
        inputs:
          - name: workspace
            resource: source-repo
      - name: build-scaffold-web
        taskRef:
          name: build-scaffold-web
        runAfter: [scaffold-unit-tests]
        resources:
          source-repo
        params:
          - name: pathToDockerfile
            value: Dockerfile
          - name: pathToContext
            value: ./workspace/workspace/examples/microservice
        resources:
          source-repo
        inputs:
          - name: workspace
```



**buildah**  
tekton | 4 | Repository  
Buildah task builds source into a container image and then pushes it to a container registry.  
0.2 (latest) | Install

**Description** YAML

**Buildah**  
This Task builds source into a container image using Project Atomic's **Buildah** build tool. It uses Buildah's support for building from **Dockerfiles**, using its **buildah bud** command. This command executes the directives in the **Dockerfile** to assemble a container image, then pushes that image to a container registry.

**Install the Task**  
kubect apply -f https://raw.githubusercontent.com/tektoncd/catalog/master/task/buildah/0.2/buildah.yaml

**Parameters**

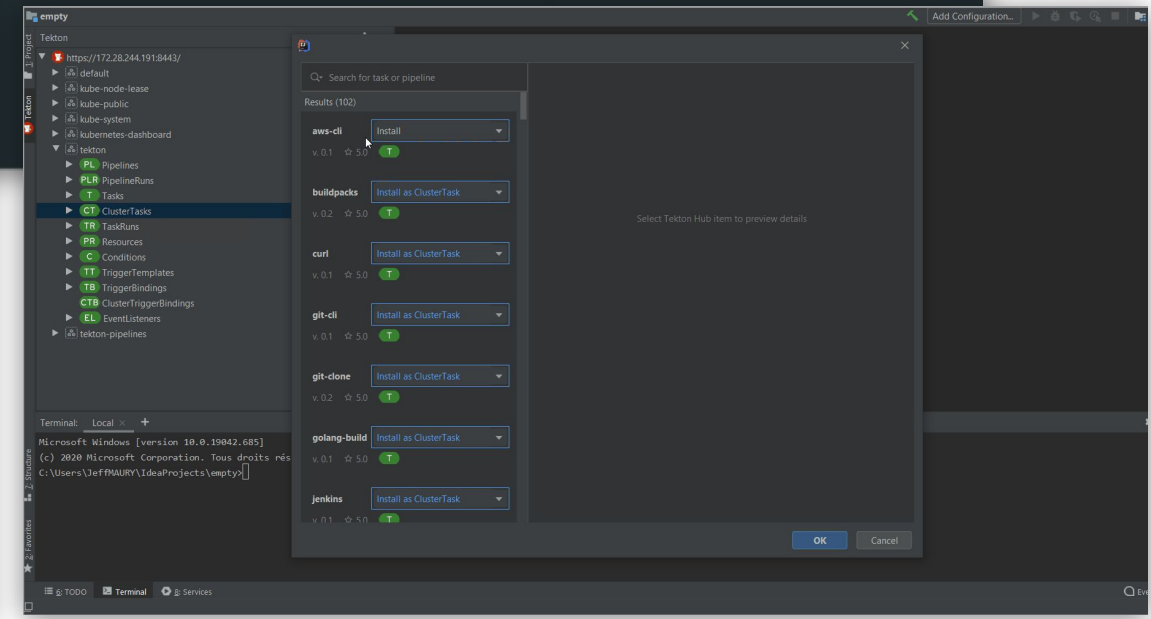
- IMAGE**: The name (reference) of the image to build.
- BUILDER\_IMAGE**: The name of the image containing the Buildah tool. See note below. (default: quay.io/buildah/stable:v1.7.0)
- DOCKERFILE**: The path to the Dockerfile to execute (default: ./Dockerfile)
- CONTEXT**: Path to the directory to use as context (default: .)
- TLSVERIFY**: Verify the TLS on the registry endpoint (for push/bull to a non-TLS registry) (default: true)
- FORMAT**: The format of the built container, oci or docker (default: oci)
- BUILD\_EXTRA\_ARGS**: Extra parameters passed for the build command when building images. (default: "")
- PUSH\_EXTRA\_ARGS**: Extra parameters passed for the push command when pushing images. (default: "")

**Workspaces**

- source**: A **Workspace** containing the source to build.
- selectorDir**: An **optional Workspace** containing your custom SSL certificates to connect to the registry. Buildah will look for files ending with **\*.cert**, **\*key** into this workspace. See [this sample](#) for a complete example on how to use it with OpenShift internal registry.

**Usage**  
This Task runs the Task to fetch a Git repo, and build and push a container image using Buildah.

```
apiVersion: tekton.dev/v1beta1
kind: TaskRun
metadata:
  name: buildah-build-my-repo
spec:
```



Search for task or pipeline

Results (102)

Task/Pipeline	Version	Status	Action
aws-cli	v.0.1	5.0	Install
buildpacks	v.0.2	5.0	Install as ClusterTask
curl	v.0.1	5.0	Install as ClusterTask
git-cli	v.0.1	5.0	Install as ClusterTask
git-clone	v.0.2	5.0	Install as ClusterTask
golang-build	v.0.1	5.0	Install as ClusterTask
jenkins	v.0.1	5.0	Install as ClusterTask

Select Tekton Hub item to preview details

OK Cancel

# OpenShift Pipelines Roadmap

## OpenShift Pipelines 1.3 (TP)

- Proxy support
- Unprivileged pipelines
- TLS support in trigger EventListener
- Tekton CLI - search and install from Tekton Hub
- VS Code - Tekton Hub integration
- IntelliJ - Tekton Hub integration
- Console - pipeline metrics and events

Q1 2021

Q2 2021

## OpenShift Pipelines 1.4 - GA

- Pipeline logs aggregation in cluster logging
- Enable TLS for EventListeners pods
- Automatic proxy env vars on TaskRuns
- ClusterTriggerBindings for BitBucket and GitLab

Q3 2021

## OpenShift Pipelines 1.5

- Pipeline as code
- Auto-pruning pipelineruns and taskruns
- OpenShift sandboxed containers (kata) support
- Jenkins to Tekton migration guide

2H 2021

## OpenShift Pipelines 1.6+

- Managed OpenShift GitOps on OSD
- Unprivileged builds in pipelines
- Approval gates in pipelines
- Auto workspace configs
- Default security tasks in pipeline templates
- In-cluster Tekton Hub and catalog
- Admin metrics dashboards
- Pipeline artifact archiving

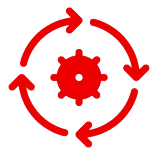


# OpenShift GitOps

Declarative GitOps for multi-cluster  
continuous delivery



# OpenShift GitOps



## Multi-cluster config management

Declaratively manage cluster and application configurations across multi-cluster OpenShift and Kubernetes infrastructure with Argo CD



## Automated Argo CD install and upgrade

Automated install, configurations and upgrade of Argo CD through OperatorHub



## Opinionated GitOps bootstrapping

Bootstrap end-to-end GitOps workflows for application delivery using Argo CD and Tekton with GitOps Application Manager CLI

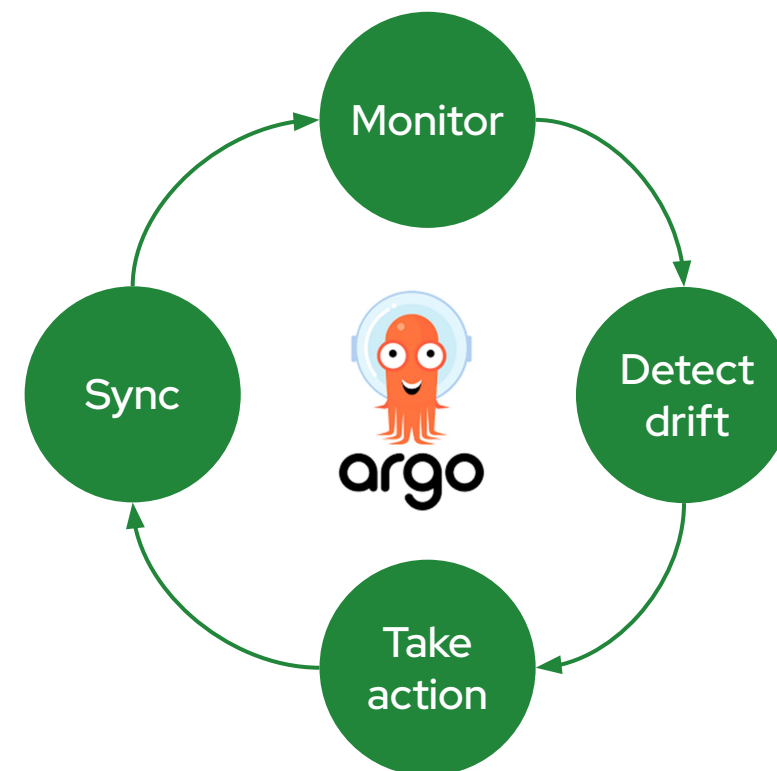


## Deployments and environments insights

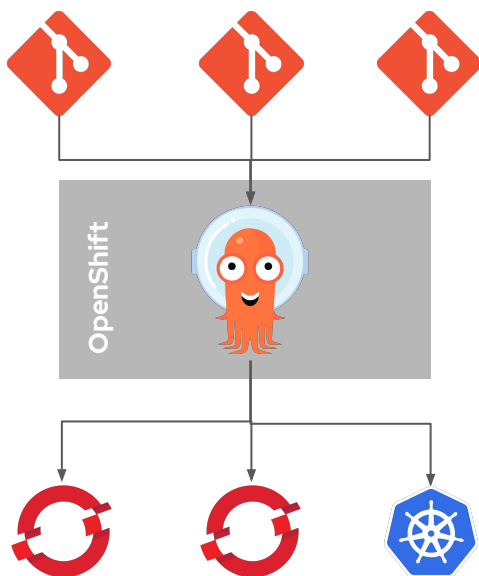
Visibility into application deployments across environments and the history of deployments in the OpenShift Console

# Argo CD

- Cluster and application configuration versioned in Git
- Automatically syncs configuration from Git to clusters
- Drift detection, visualization and correction
- Granular control over sync order for complex rollouts
- Rollback and rollforward to any Git commit
- Manifest templating support (Helm, Kustomize, etc)
- Visual insight into sync status and history

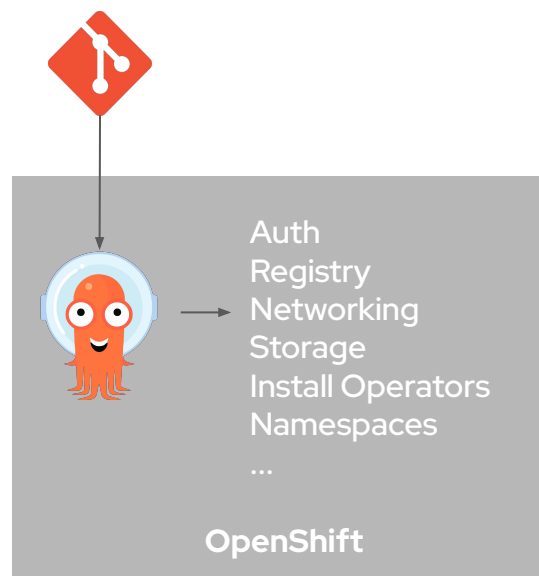


# Flexible Deployment Strategies



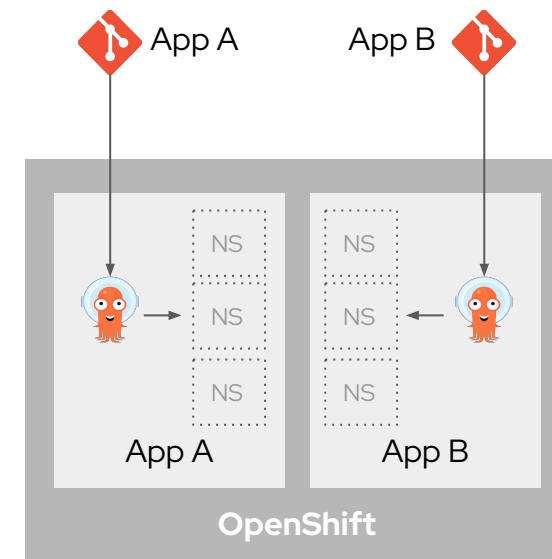
## Central Hub (Push)

A central Argo CD pushes Git repository content to remote OpenShift and Kubernetes clusters



## Cluster Scoped (Pull)

A cluster-scope Argo CD pulls cluster service configurations into the OpenShift cluster



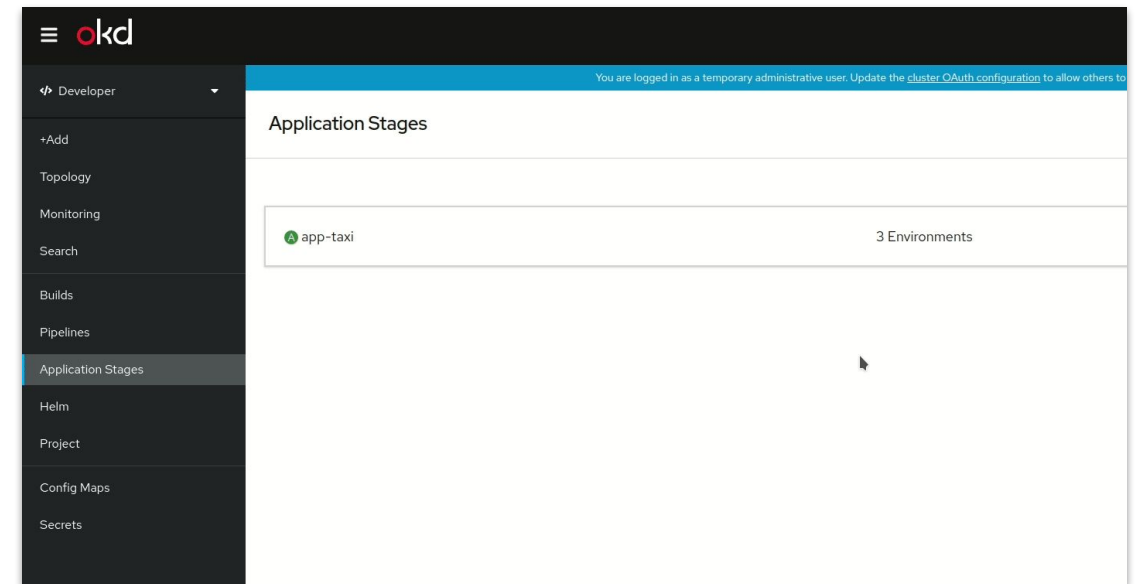
## Application Scoped (Pull)

An application scoped Argo CD pulls application deployment and configurations into app namespaces

# GitOps Application Manager CLI

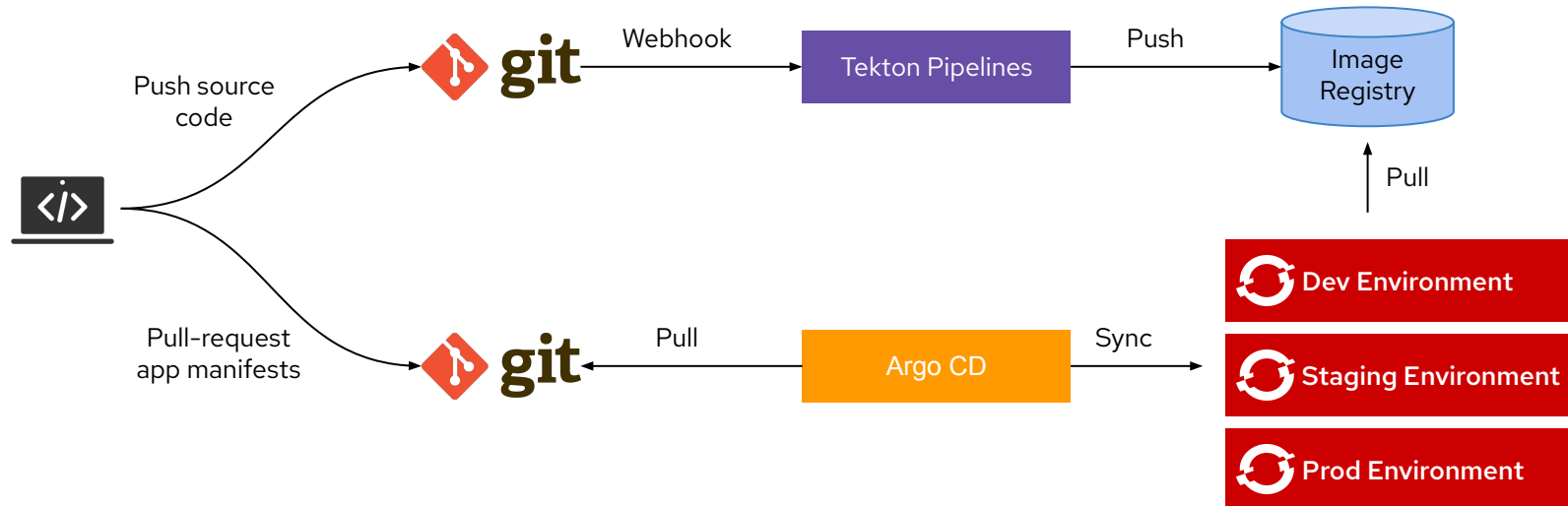
- Bootstraps Git repos for GitOps
- Configures deployment environments
- Configures webhooks for Tekton Pipelines for CI
- Configures Argo CD for deployment to environments
- Kustomize for environment-specific configs
- Integration with secret managers

```
$ kam bootstrap  
$ kam environment add stage
```



# GitOps Application Manager CLI

```
$ kam bootstrap
```



# OpenShift GitOps Roadmap

## OpenShift GitOps 1.0 - Tech Preview

- Pre-configured Argo CD for cluster configs
- Argo DD in air-gapped clusters
- Argo CD proxy config support
- Argo CD deployment guide
- Argo CD metrics in cluster Prometheus
- Out of sync alerts in cluster AlertManager

Q1 2021

Q2 2021

## OpenShift GitOps 1.1 - GA

- Argo CD logs aggregated in cluster logging stack
- Guidance on auth integrations with OpenShift
- Dynamic generation of Applications (ApplicationSets)

## OpenShift GitOps 1.2

- Cluster Argo CD uses OpenShift auth
- kam - simplified Git repo layout
- Add custom Argo CD plugins and tools
- Secret management guidance
- Dev Console - application environments views

Q3 2021

Q4 2021

## OpenShift GitOps 1.3+

- Namespaced Argo CD uses OpenShift auth
- Managed OpenShift GitOps on OSD
- ApplicationSets integration with RHACM
- Health status for OpenShift resources
- Multi-tenant Argo CD control plane
- Helm Charts in kam

# 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)



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



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



[twitter.com/RedHat](https://twitter.com/RedHat)