Continuous Delivery and GitOps on OpenShift

Christian Hernandez
Technical Marketing Manager
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

- Git is the single source of truth
- Treat everything as code
- Operations through Git workflows
GitOps Workflow
a declarative approach to application delivery

What you want
(desired state)

What you have
(current state)
Why GitOps?

**Standard Workflow**
Familiar tools and Git workflows from application development teams

**Visibility and Audit**
Capturing and tracing any change to clusters through Git history

**Enhanced Security**
Review changes beforehand, detect configuration drifts, and take action

**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

- Source Git Repository
- CI
- Image Registry
- Pull Request
- Config Git Repository
- CD
- Push Pull
- Kubernetes
- Monitor
- Deploy
- Detect drift
- Take action
Continuous Integration & Continuous Delivery

**Build**
- **OpenShift Build**
  - Automate building container images using Kubernetes tools

**Test**
- **OpenShift Pipelines**
  - Kubernetes-native on-demand delivery pipelines

**Security Checks**

**Release**

**Deploy Stage**

**Deploy Prod**
- **OpenShift GitOps**
  - Declarative GitOps for multi-cluster continuous delivery

Ecosystem Integrations

GitHub, GitLab, Azure DevOps, Travis CI
The GitOps Application Delivery Model on OpenShift
OpenShift Builds

Automate building container images using Kubernetes tools
OpenShift Builds

**Kubernatives-native image build**
A Kubernative-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

<table>
<thead>
<tr>
<th>Cloud-Native Buildpacks</th>
<th>Source-to-Image (S2I)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>kind</strong>: Build</td>
<td><strong>kind</strong>: Build</td>
</tr>
<tr>
<td><strong>metadata</strong>:</td>
<td><strong>metadata</strong>:</td>
</tr>
<tr>
<td><strong>name</strong>: myapp-buildpack</td>
<td><strong>name</strong>: myapp-s2i</td>
</tr>
<tr>
<td><strong>spec</strong>:</td>
<td><strong>spec</strong>:</td>
</tr>
<tr>
<td><strong>source</strong>:</td>
<td><strong>source</strong>:</td>
</tr>
<tr>
<td><strong>strategy</strong>:</td>
<td><strong>strategy</strong>:</td>
</tr>
<tr>
<td><strong>name</strong>: buildpacks-v3</td>
<td><strong>name</strong>: source-to-image</td>
</tr>
<tr>
<td><strong>builder</strong>:</td>
<td><strong>builder</strong>:</td>
</tr>
<tr>
<td><strong>image</strong>: paketobuildpacks/builder:full</td>
<td><strong>image</strong>: registry.redhat.io/openjdk/openjdk-11-rhel8</td>
</tr>
<tr>
<td><strong>output</strong>:</td>
<td><strong>output</strong>:</td>
</tr>
<tr>
<td><strong>image</strong>: quay.io/myorg/myapp:v1</td>
<td><strong>image</strong>: quay.io/myorg/myapp:v1</td>
</tr>
<tr>
<td></td>
<td><strong>runtime</strong>:</td>
</tr>
<tr>
<td></td>
<td><strong>image</strong>: docker.io/openjdk:11-jre-slim</td>
</tr>
</tbody>
</table>
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?

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

Powered by TEKTON
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
    ...

OPENSHIFT PIPELINES
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.
Tekton Concepts: Pipeline

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

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

Powered by

Red Hat
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
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
facebook.com/redhatinc
youtube.com/user/RedHatVideos
twitter.com/RedHat