

# Dekorate

A set of tools for generating Kubernetes manifests

Aurea Munoz @auritamh  
Software Engineer at Red Hat



# Agenda

---

- Who am I?
- Cloud native era
- Kubernetes manifests vs Kubernetes resources
- What is Dekorater?
- Generation Process Pipeline with Dekorater
- Demo
- More Dekorater features
- Dekorater resources
- Questions

# Who am I?



Auri Muñoz  
Senior Software engineer  
auri@redhat.com  
Red Hat - Snowdrop team



# Cloud native & Container platforms

The background of the slide is a photograph of the Eiffel Tower in Paris, France, viewed from across the Seine river. The entire image is covered with a semi-transparent teal or cyan color overlay, which serves as a background for the white text.

# Kubernetes manifest & Kubernetes resource



## Kubernetes manifest

```

apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: demo
    app.kubernetes.io/version: 0.0.1-SNAPSHOT
  name: demo
spec:
  ports:
    - name: http
      port: 8080
      targetPort: 8080
  selector:
    app.kubernetes.io/name: demo
    app.kubernetes.io/version: 0.0.1-SNAPSHOT
  type: ClusterIP
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app.kubernetes.io/version: 0.0.1-SNAPSHOT
    app.kubernetes.io/name: demo
  name: demo
spec:
  replicas: 4
  selector:
    matchLabels:
      app.kubernetes.io/version: 0.0.1-SNAPSHOT
      app.kubernetes.io/name: demo
  template:
    metadata:
      labels:
        app.kubernetes.io/version: 0.0.1-SNAPSHOT
        app.kubernetes.io/name: demo
    spec:
      containers:
        - env:
            - name: KUBERNETES_NAMESPACE
              valueFrom:
                fieldRef:
                  fieldPath: metadata.namespace
            image: amunozhe/demo:0.0.1-SNAPSHOT
            imagePullPolicy: IfNotPresent
            name: demo
            ports:
              - containerPort: 8080
              name: http
              protocol: TCP

```

## Service

## Deployment

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    app.dekorate.io/vcs-url: <unknown>
    app.dekorate.io/commit-id: 0705224b727a8e2954ded381d78e3acc3b77bf7
  labels:
    app.kubernetes.io/name: demo
    app.kubernetes.io/version: 0.0.1-SNAPSHOT
  name: demo
spec:
  rules:
    - host: sb-demo.127.0.0.1.nip.io
      http:
        paths:
          - backend:
              service:
                name: demo
                port:
                  name: http
              path: /
              pathType: Prefix

```

## Ingress

## Kubernetes Resources



# What is Dekorate?

Dekorate is a collection of annotations and processors that generates Kubernetes manifests during the compilation of your application

- Works for all build tools and all jvm languages
- Allows fine grained customization.
- Can be used as a library, a Kubernetes manifest manipulation framework.
- The simplest out of the box experience: framework detection



# Generation Process Pipeline with Dekorater



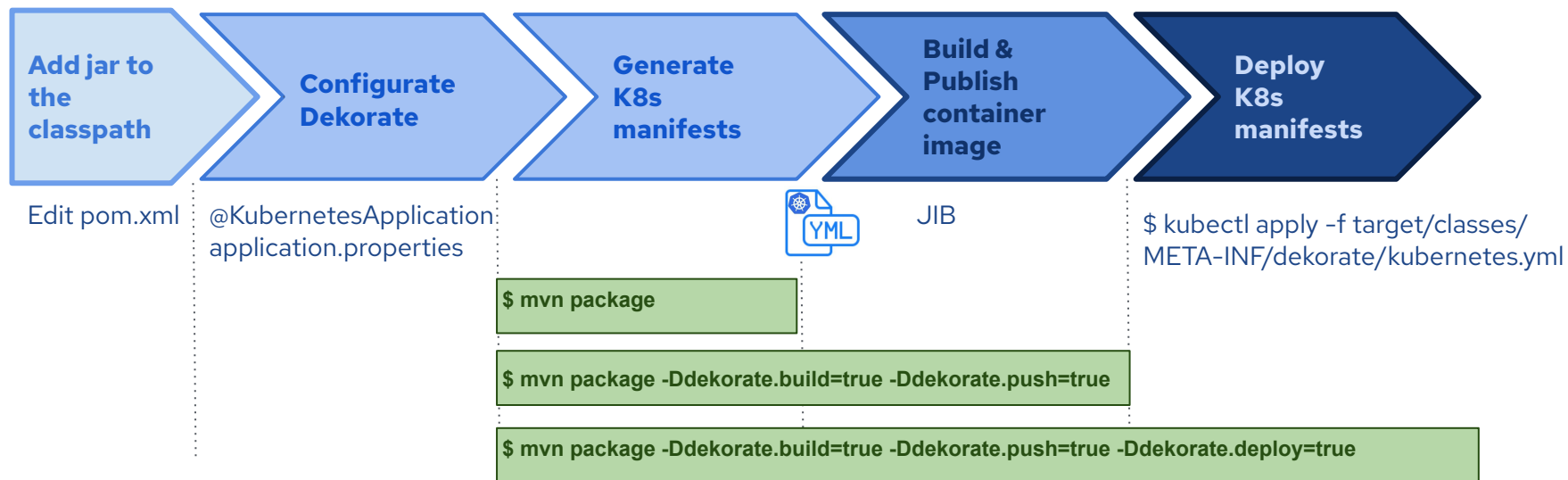
# Live Demo

Using Dekorater in a Spring Boot application





# Demo Summary



# Dekorate features

- Kubernetes
- OpenShift
- Knative
- Tekton
- Operator custom resources
  - Prometheus
  - Jaeger
- Spring
- Quarkus
- Micronaut
- Frameworkless
- ...more



# Dekorate resources



<https://github.com/dekorateio/dekorate>



[Complete reference of the properties](#)



[Dekorate examples](#)



<https://snowdrop/zulipchat.com>



[Using Dekorate for frameworkless Java applications](#)



**Let's Dekorate!**  
**Merci !!**

