

Microservices – from Docker to Kubernetes

Kalin Staykov

December 2020

docker-compose

common beginning

Quick and easy to use.

But not flexible and hard to scale.

```
version: "3.8"

services:
  db:
    image: postgres
    environment:
      - POSTGRES_DB=postgres
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres
  web:
    build: .
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - ./code
    ports:
      - "8000:8000"
    depends_on:
      - db
```

Kubernetes

common production hosting

Solves the prod use case in a great way.

But it's complex and needs a lot of attention during development which is distracting.

Replica sets
Services
Stateful sets
Persistent volumes
Policy Pods
Deployments
Ephemeral volumes
Resource limits
Yamls... so many yamls.

K8s is nice but I want to focus on development

Good foundation goes a long way.

Meet Skaffold and Kustomize.



Kustomize

Patch yaml definitions for different use cases

Base config with the standard definition.

Patch with the specific use case.

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

match

kustomization.yaml

```
commonAnnotations:
  imageregistry: hub.docker.com
resources:
- deployment.yaml
patchesStrategicMerge:
- overlay.yaml
```

overlay.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 10
```

Example: Apply common annotations to all Kube deployment files and apply an overlay to a specific deployment

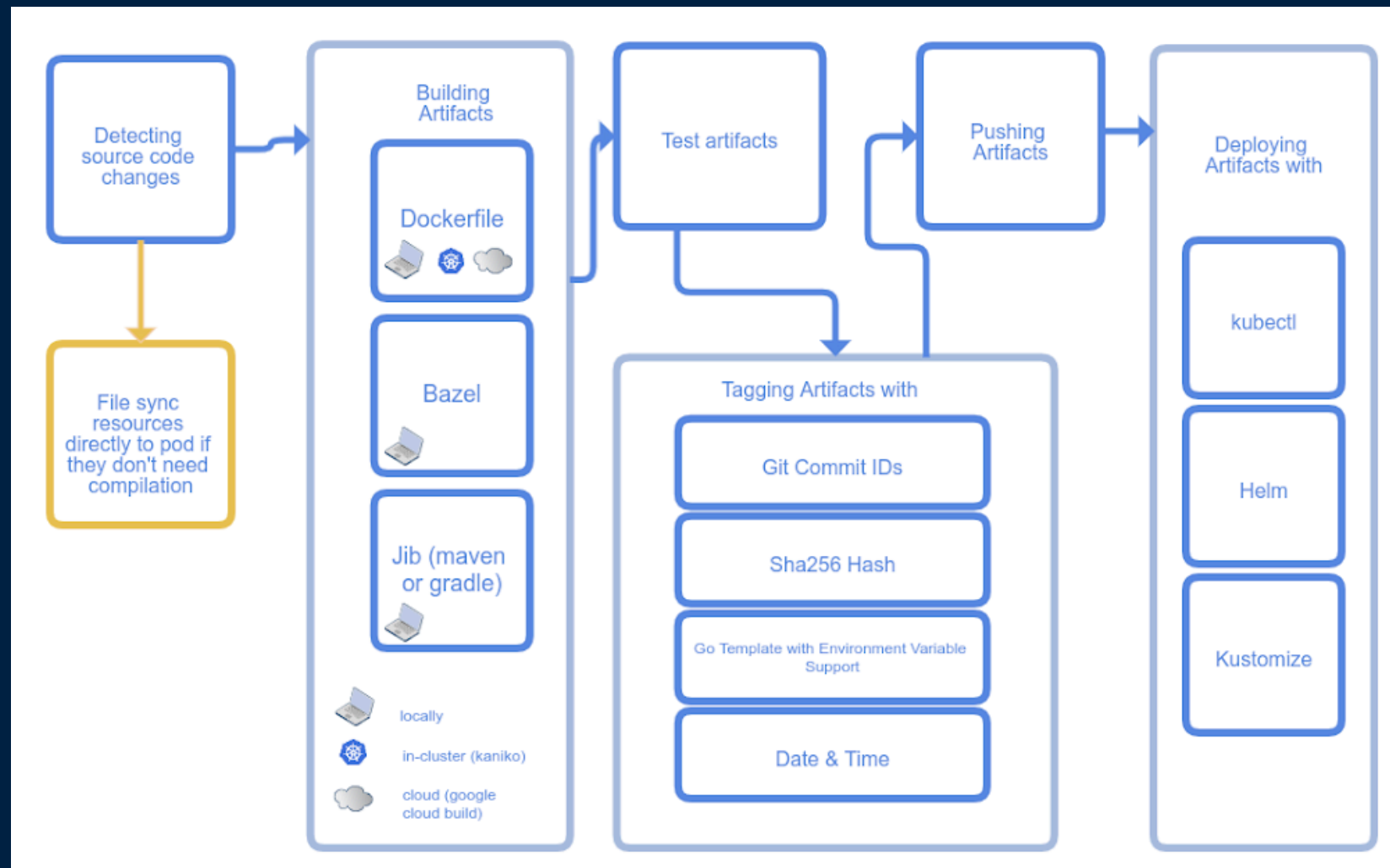
(generated) deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  annotations:
    imageregistry: hub.docker.com
  labels:
    app: nginx
spec:
  replicas: 10
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      annotations:
        imageregistry: hub.docker.com
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:1.14.2
        name: nginx
        ports:
        - containerPort: 80
```

Scaffold

Build, test and deploy with ease

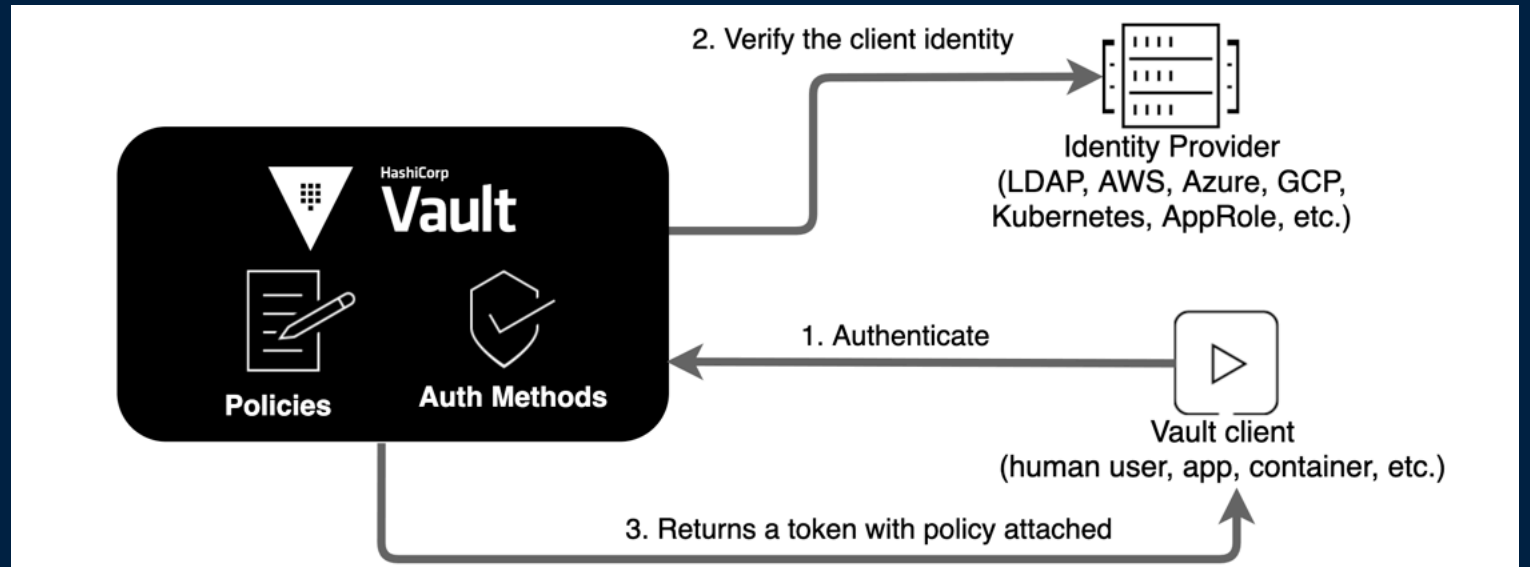
Like docker-compose
but much more
powerful.



Vault

Not just for secrets..

Vault can contain json files with sensitive information.
Useful for the entire application configuration.



Hands-on session

Before we begin



Develop on both local and remote Kubernetes cluster with hot reload support and debugging capabilities.

Automated configuration using Hashicorp's Vault.

Scaffold is not the only option

The Kubernetes eco system is vast

<https://landscape.cncf.io/>



Q&A



Thank You

Please email any questions to kstaykov@vmware.com