Lab 28 : CICD

mkdir /mnt/data2

mkdir /mnt/data1

Create a yaml file from below code

apiVersion: v1

kind: PersistentVolume

metadata:

name: pv-volume-2

labels:

type: local

spec:

storageClassName: manual

capacity:

storage: 1Gi

accessModes:

- ReadWriteOnce

hostPath:

path: "/mnt/data2"

kubectl create -f <name of yaml file>

apiVersion: v1

kind: PersistentVolume

metadata:

name: pv-volume-2

labels:

type: local

spec:

storageClassName: manual

capacity:

storage: 1Gi

accessModes:

- ReadWriteOnce

hostPath:

path: "/mnt/data1"

#https://medium.com/atlantbh/implementing-ci-cd-pipeline-using-argo-workflows-and-argo-events-6417dd157566

apiVersion: argoproj.io/v1alpha1

kind: WorkflowTemplate

metadata:

name: cicd-workflow-template

namespace: argo

annotations:

workflows.argoproj.io/description: |

This workflow is implementing CICD for simple GO API.

spec:

serviceAccountName: training

arguments:

parameters:

- name: branch

value: main

- name: dockerRepo

value: docker.io/amitow/argo-workflows:1.0.0

entrypoint: main

podGC:

strategy: OnPodSuccess

volumeClaimTemplates:

- metadata:

name: work

spec:

storageClassName: manual # Specify your StorageClass name here

accessModes: [ ReadWriteOnce ]

resources:

requests:

storage: 64Mi

templates:

- name: main

dag:

tasks:

- name: clone

template: clone

- name: deps

template: deps

dependencies:

- clone

- name: build

template: build

dependencies:

- deps

- name: docker-build

template: docker-build

dependencies:

- build

- name: create-config

template: create-config-k8s

dependencies:

- docker-build

- name: create-secret

template: create-secret-k8s

dependencies:

- docker-build

- name: deploy

template: deploy

dependencies:

- create-config

- create-secret

- name: clone

container:

volumeMounts:

- mountPath: /go/src/github.com/go-api

name: work

subPath: src

- mountPath: /go/pkg/mod

name: work

subPath: GOMODCACHE

- mountPath: /root/.cache/go-build

name: work

subPath: GOCACHE

image: golang:1.18

workingDir: /go/src/github.com/go-api

command: [ sh, -euxc ]

args:

- |

git clone -v -b "{{workflow.parameters.branch}}" --single-branch --depth 1 https://github.com/adnanselimovic-abh/go-api.git .

- name: deps

container:

image: golang:1.18

volumeMounts:

- mountPath: /go/src/github.com/go-api

name: work

subPath: src

- mountPath: /go/pkg/mod

name: work

subPath: GOMODCACHE

- mountPath: /root/.cache/go-build

name: work

subPath: GOCACHE

workingDir: /go/src/github.com/go-api

command: [ sh, -xuce ]

args:

- |

go mod tidy

- name: build

container:

image: golang:1.18

volumeMounts:

- mountPath: /go/src/github.com/go-api

name: work

subPath: src

- mountPath: /go/pkg/mod

name: work

subPath: GOMODCACHE

- mountPath: /root/.cache/go-build

name: work

subPath: GOCACHE

workingDir: /go/src/github.com/go-api

command: [ sh, -xuce ]

args:

- |

CGO\_ENABLED=0 go build main.go

- name: docker-build

volumes:

- name: docker-config

secret:

secretName: docker-config

container:

image: moby/buildkit:v0.9.3-rootless

volumeMounts:

- mountPath: /go/src/github.com/go-api

name: work

subPath: src

- name: docker-config

mountPath: /.docker

workingDir: /go/src/github.com/go-api

env:

- name: BUILDKITD\_FLAGS

value: --oci-worker-no-process-sandbox

- name: DOCKER\_CONFIG

value: /.docker

command:

- buildctl-daemonless.sh

args:

- build

- --frontend

- dockerfile.v0

- --local

- context=.

- --local

- dockerfile=docker/

- --output

- type=image,name={{workflow.parameters.dockerRepo}},push=true

securityContext:

privileged: true

- name: create-config-k8s

resource:

action: apply

manifest: |

apiVersion: v1

kind: ConfigMap

metadata:

name: go-api-config

namespace: go-api

data:

branch: {{workflow.parameters.branch}}

- name: create-secret-k8s

resource:

action: apply

manifest: |

apiVersion: v1

kind: Secret

type: Opaque

metadata:

name: go-api-secret

namespace: go-api

data:

username: YWRtaW4=

- name: deploy

container:

image: dtzar/helm-kubectl:3.9

volumeMounts:

- mountPath: /go/src/github.com/go-api-helm

name: work

subPath: src

workingDir: /go/src/github.com/go-api-helm

command: [ sh, -xuce ]

args:

- |

[ ! -d "go-api-helm" ] && git clone -v -b "{{workflow.parameters.branch}}" --single-branch --depth 1 https://github.com/adnanselimovic-abh/go-api-helm.git

cd go-api-helm

helm upgrade --install atlantbh . --namespace go-api --set image.tag=1.0.0