Project Lab ci

apiVersion: argoproj.io/v1alpha1

kind: Workflow

metadata:

  generateName: ci-example-

  namespace: argo

spec:

  # entrypoint is the name of the template used as the starting point of the workflow

  entrypoint: ci-example

  serviceAccountName: training

  # the 'ci-example' template accepts a parameter 'revision', with a default of 'cfe12d6'.

  # this can be overridden via argo CLI (e.g. `argo submit ci.yaml -p revision=0dea2d0`)

  arguments:

    parameters:

    - name: revision

      value: cfe12d6

  # a temporary volume, named workdir, will be used as a working directory

  # for this workflow. This volume is passed around from step to step.

  volumeClaimTemplates:

  - metadata:

      name: workdir

    spec:

      accessModes: [ "ReadWriteOnce" ]

      storageClassName: manual

      resources:

        requests:

          storage: 1Gi

  templates:

  - name: ci-example

    inputs:

      parameters:

      - name: revision

    steps:

    - - name: build

        template: build-golang-example

        arguments:

          parameters:

          - name: revision

            value: "{{inputs.parameters.revision}}"

    # the test step expands into three parallel steps running

    # different operating system images. each mounts the workdir

    # and runs the compiled binary from the build step.

    - - name: test

        template: run-hello

        arguments:

          parameters:

          - name: os-image

            value: "{{item.image}}:{{item.tag}}"

        withItems:

        - { image: 'debian', tag: '9.1' }

        - { image: 'alpine', tag: '3.6' }

        - { image: 'ubuntu', tag: '17.10' }

  - name: build-golang-example

    inputs:

      parameters:

      - name: revision

      artifacts:

      - name: code

        path: /go/src/github.com/golang/example

        git:

          repo: https://github.com/golang/example.git

          revision: "{{inputs.parameters.revision}}"

    container:

      image: golang:1.8

      command: [sh, -c]

      args: ["

        cd /go/src/github.com/golang/example/hello &&

        git status &&

        go build -v .

      "]

      volumeMounts:

      - name: workdir

        mountPath: /go

  - name: run-hello

    inputs:

      parameters:

      - name: os-image

    container:

      image: "{{inputs.parameters.os-image}}"

      command: [sh, -c]

      args: ["

        uname -a ;

        cat /etc/os-release ;

        /go/src/github.com/golang/example/hello/hello

      "]

      volumeMounts:

      - name: workdir

        mountPath: /go