Lab Kubernetes Jobs

# in a workflow. The resource template type accepts any k8s manifest

# (including CRDs) and can perform any `kubectl` action against it (e.g. create,

# apply, delete, patch).

apiVersion: argoproj.io/v1alpha1

kind: Workflow

metadata:

  generateName: k8s-jobs-

  namespace: argo

spec:

  entrypoint: pi-tmpl

  serviceAccountName: training

  templates:

  - name: pi-tmpl

    resource:                   # indicates that this is a resource template

      action: create            # can be any kubectl action (e.g. create, delete, apply, patch)

      # The successCondition and failureCondition are optional expressions.

      # If failureCondition is true, the step is considered failed.

      # If successCondition is true, the step is considered successful.

      # They use kubernetes label selection syntax and can be applied against any field

      # of the resource (not just labels). Multiple AND conditions can be represented by comma

      # delimited expressions.

      # For more details: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/

      successCondition: status.succeeded > 0

      failureCondition: status.failed > 3

      manifest: |               #put your kubernetes spec here

        apiVersion: batch/v1

        kind: Job

        metadata:

          generateName: pi-job-

        spec:

          template:

            metadata:

              name: pi

            spec:

              containers:

              - name: pi

                image: perl

                command: ["perl",  "-Mbignum=bpi", "-wle", "print bpi(2000)"]

              restartPolicy: Never

          backoffLimit: 4

# Second flow

apiVersion: argoproj.io/v1alpha1

kind: Workflow

metadata:

generateName: k8s-patch-

namespace: argo

spec:

entrypoint: cront-tmpl

serviceAccountName: training

templates:

- name: cront-tmpl

resource:

action: patch

mergeStrategy: merge # Must be one of [strategic merge json]

manifest: |

apiVersion: "stable.example.com/v1"

kind: CronTab

spec:

cronSpec: "\* \* \* \* \*/10"

image: my-awesome-cron-image