Lab 22

apiVersion: argoproj.io/v1alpha1

kind: Workflow

metadata:

generateName: container-set-template-

namespace: argo

spec:

entrypoint: main

serviceAccountName: training

templates:

- name: main

volumes:

- name: workspace

emptyDir: { }

containerSet:

volumeMounts:

- mountPath: /workspace

name: workspace

containers:

- name: a

image: argoproj/argosay:v2

command: [sh, -c]

args: ["echo 'a: hello world' >> /workspace/message"]

- name: b

image: argoproj/argosay:v2

command: [sh, -c]

args: ["echo 'b: hello world' >> /workspace/message"]

- name: main

image: argoproj/argosay:v2

command: [sh, -c]

args: ["echo 'main: hello world' >> /workspace/message"]

dependencies:

- a

- b

outputs:

parameters:

- name: message

valueFrom:

path: /workspace/message

ContiainerSet Retry

apiVersion: argoproj.io/v1alpha1

kind: Workflow

metadata:

name: containerset-with-retrystrategy

namespace: argo

annotations:

workflows.argoproj.io/description: |

This workflow creates a container set with a retryStrategy.

spec:

entrypoint: containerset-retrystrategy-example

serviceAccountName: training

templates:

- name: containerset-retrystrategy-example

containerSet:

retryStrategy:

retries: "10" # if fails, retry at most ten times

duration: 30s # retry for at most 30s

containers:

# this container completes successfully, so it won't be retried.

- name: success

image: python:alpine3.6

command:

- python

- -c

args:

- |

print("hi")

# if fails, it will retry at most ten times.

- name: fail-retry

image: python:alpine3.6

command: ["python", -c]

# fail with a 66% probability

args: ["import random; import sys; exit\_code = random.choice([0, 1, 1]); sys.exit(exit\_code)"]