Lab 21

Try using UI

apiVersion: argoproj.io/v1alpha1

kind: Workflow

metadata:

generateName: http-template-

namespace: argo

spec:

entrypoint: main

serviceAccountName: training # Add the service account here for triggering the Workflow

templates:

- name: main

steps:

- - name: get-google-homepage

template: http

arguments:

parameters: [{name: url, value: "https://www.google.com"}]

- name: http

inputs:

parameters:

- name: url

http:

timeoutSeconds: 20 # Default 30

url: "{{inputs.parameters.url}}"

method: "GET" # Default GET

headers:

- name: "x-header-name"

value: "test-value"

# Template will succeed if evaluated to true, otherwise will fail

# Available variables:

# request.body: string, the request body

# request.headers: map[string][]string, the request headers

# response.url: string, the request url

# response.method: string, the request method

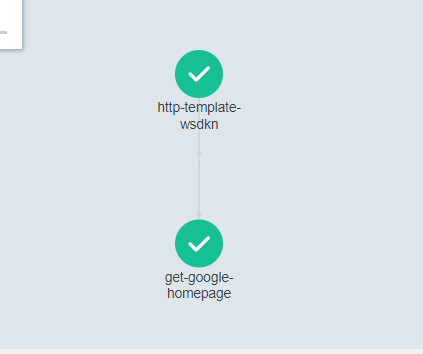
# response.statusCode: int, the response status code

# response.body: string, the response body

# response.headers: map[string][]string, the response headers

successCondition: "response.body contains \"google\"" # available since v3.3

body: "test body" # Change request body



Examples: Try above yaml file with cli as well

# Submit multiple workflows from files:

argo submit my-wf.yaml

# Submit and wait for completion:

argo submit --wait my-wf.yaml

# Submit and watch until completion:

argo submit --watch my-wf.yaml

# Submit and tail logs until completion:

argo submit --log my-wf.yaml

# Submit a single workflow from an existing resource

argo submit --from cronwf/my-cron-wf