Lab 23

apiVersion: argoproj.io/v1alpha1

kind: Workflow

metadata:

generateName: diamond-dag-

namespace: argo

spec:

entrypoint: main

serviceAccountName: training

templates:

- name: main

dag:

tasks:

- name: a

template: task-a

resources:

limits:

cpu: "1000m" # 1000 millicores = 1 CPU

- name: b

dependencies: [a]

template: task-b

resources:

limits:

cpu: "2000m" # 2000 millicores = 2 CPUs

- name: c

dependencies: [a]

template: task-c

resources:

limits:

cpu: "1000m" # 1000 millicores = 1 CPU

- name: d

dependencies: [b, c]

template: task-d

resources:

limits:

cpu: "1000m" # 1000 millicores = 1 CPU

- name: task-a

container:

image: argoproj/argosay:v2

command: [sh, -c]

args: ["echo 'a: hello world'"]

- name: task-b

container:

image: argoproj/argosay:v2

command: [sh, -c]

args: ["echo 'b: hello world'"]

- name: task-c

container:

image: argoproj/argosay:v2

command: [sh, -c]

args: ["echo 'c: hello world'"]

- name: task-d

container:

image: argoproj/argosay:v2

command: [sh, -c]

args: ["echo 'd: hello world'"]

Try using CLI

apiVersion: argoproj.io/v1alpha1

kind: Workflow

metadata:

generateName: lopsided-requests-

namespace: argo

spec:

entrypoint: main

serviceAccountName: training

templates:

- name: main

dag:

tasks:

- name: a

template: task-a

resources:

limits:

cpu: "100m" # 100 millicores = 0.1 CPU

memory: "1Mi" # 1 Mebibyte

retryStrategy:

limit: 3

retryPolicy: Always

duration: "2m" # 2 minutes

- name: b

dependencies: [a]

template: task-b

resources:

limits:

cpu: "8000m" # 8000 millicores = 8 CPUs

memory: "100Gi" # 100 Gibibytes

nvidia.com/gpu: "200m" # 200 millicores = 0.2 GPU

retryStrategy:

limit: 1

retryPolicy: Never

duration: "1m" # 1 minute

- name: task-a

container:

image: argoproj/argosay:v2

command: [sh, -c]

args: ["echo 'a: hello world' && sleep 5"]

- name: task-b

container:

image: argoproj/argosay:v2

command: [sh, -c]

args: ["echo 'b: hello world' && sleep 2"]