apiVersion: tekton.dev/v1

kind: Pipeline

metadata:

name: py-echo-ci-v3

spec:

params:

- name: repo\_url

type: string

- name: revision

type: string

workspaces:

- name: source

tasks:

# 1) Clone the repo into the workspace PVC

- name: clone

params:

- name: repo\_url

value: $(params.repo\_url)

- name: revision

value: $(params.revision)

workspaces:

- name: output

workspace: source

taskSpec:

params:

- name: repo\_url

- name: revision

workspaces:

- name: output

steps:

- name: git-clone

image: alpine/git:2.44

workingDir: $(workspaces.output.path)

script: |

#!/bin/sh

set -eu

echo "FPRINT: [clone] py-echo-ci-v3 | $(date -Iseconds)"

echo "Cloning $(params.repo\_url) @ $(params.revision) ..."

git clone --depth 1 --branch "$(params.revision)" "$(params.repo\_url)" .

echo "HEAD commit:" && git rev-parse HEAD

echo "Repo root after clone:"

ls -al

# 2) Echo-only step (no installs)

- name: echo-step

runAfter: [clone]

workspaces:

- name: src

workspace: source

taskSpec:

workspaces:

- name: src

steps:

- name: debug-echo

image: python:3.11-slim

workingDir: $(workspaces.src.path)

script: |

#!/bin/sh

set -eu

echo "FPRINT: [echo-step] py-echo-ci-v3 | $(date -Iseconds)"

echo "PWD=$(pwd)"

echo "WORKSPACE path: $(workspaces.src.path)"

echo "Files here:" && ls -al

echo "Message: Hello from Tekton echo step!"

\\\

apiVersion: tekton.dev/v1

kind: PipelineRun

metadata:

generateName: py-echo-ci-v3-

spec:

pipelineRef:

name: py-echo-ci-v3

params:

- name: repo\_url

value: https://github.com/<YOUR\_ORG\_OR\_USER>/<YOUR\_REPO>.git

- name: revision

value: <YOUR\_BRANCH> # e.g., ci-clean-start

workspaces:

- name: source

volumeClaimTemplate:

spec:

accessModes: ["ReadWriteOnce"]

resources:

requests:

storage: 1Gi