**pipeline.yaml**

apiVersion: tekton.dev/v1

kind: PipelineRun

metadata:

generateName: simple-echo-run-

spec:

pipelineRef:

name: simple-echo-pipeline

taskRunTemplate:

podTemplate:

env:

- name: HOME

value: /tekton/home

- name: DOCKER\_CONFIG

value: /tekton/home/.docker

params:

- name: repo\_url

value: https://github.com/CitiInternal/178144\_jedi.jedi-ml.git

- name: revision

value: ci-clean-start

workspaces:

- name: source

volumeClaimTemplate:

spec:

accessModes: ["ReadWriteOnce"]

resources:

requests:

storage: 2Gi

**pipelinerun.yaml**

apiVersion: tekton.dev/v1

kind: Pipeline

metadata:

name: simple-echo-pipeline

spec:

params:

- name: repo\_url

type: string

description: Git repo to clone

- name: revision

type: string

description: Branch or tag to checkout

workspaces:

- name: source

tasks:

- 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] simple-echo-pipeline | $(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 tree:"

find . -maxdepth 2 -type f | sort

- name: echo-step

runAfter: [clone]

workspaces:

- name: src

workspace: source

taskSpec:

workspaces:

- name: src

steps:

- name: say-hello

image: python:3.11-slim

workingDir: $(workspaces.src.path)

env:

- name: WORKSPACE

value: $(workspaces.src.path)

script: |

#!/bin/sh

set -eu

echo "FPRINT: [echo-step] simple-echo-pipeline | $(date -Iseconds)"

echo "PWD=$(pwd)"

echo "WORKSPACE=$WORKSPACE"

echo "Files here:" && ls -al

echo "✅ Hello from \*root-level\* pipeline.yaml"