apiVersion: tekton.dev/v1

kind: Pipeline

metadata:

name: py-reqs-ci

spec:

params:

- name: repo\_url

type: string

- name: revision

type: string

default: feature/ci-sandbox

workspaces:

- name: source

tasks:

# 1) Clone this branch into the workspace

- 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

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

git status

# 2) Install requirements

- name: pip-install

runAfter: [clone]

workspaces:

- name: src

workspace: source

taskSpec:

workspaces:

- name: src

steps:

- name: install

# if Docker Hub is blocked in your cluster, switch to: registry.access.redhat.com/ubi9/python-311

image: python:3.11-slim

workingDir: $(workspaces.src.path)

env:

- name: HOME

value: /tekton/home # avoids /.docker permission noise

- name: PIP\_DISABLE\_PIP\_VERSION\_CHECK

value: "1"

script: |

#!/bin/sh

set -eu

python -V

pip install --upgrade pip

if [ -f requirements.txt ]; then

pip install -r requirements.txt

else

echo "No requirements.txt in repo root."

Fi

apiVersion: tekton.dev/v1

kind: PipelineRun

metadata:

generateName: py-reqs-ci-

spec:

pipelineRef:

name: py-reqs-ci

# Make creds-init and steps use a writable HOME (silences /.docker errors)

taskRunTemplate:

podTemplate:

env:

- name: HOME

value: /tekton/home

- name: DOCKER\_CONFIG

value: /tekton/home/.docker

params:

- name: repo\_url

value: https://github.com/<YOUR\_ORG>/<YOUR\_REPO>.git # <-- put your repo URL

- name: revision

value: feature/ci-sandbox # <-- your sandbox branch

workspaces:

- name: source

volumeClaimTemplate:

spec:

accessModes: ["ReadWriteOnce"]

resources:

requests:

storage: 2Gi