**pipeline.yaml**

apiVersion: tekton.dev/v1

kind: Pipeline

metadata:

name: py-reqs-ci

spec:

params:

- name: repo\_url

type: string

- name: revision

type: string

default: ci-clean-start

workspaces:

- name: source

tasks:

# 1) Clone the repo to the 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-reqs-ci $(date -Iseconds)"

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

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

echo "HEAD: $(git rev-parse HEAD)"

echo "Repo root content:"

ls -al

# 2) Only echo & (optionally) install requirements.txt if present

- name: pip-install

runAfter: [clone]

workspaces:

- name: src

workspace: source

taskSpec:

workspaces:

- name: src

steps:

- name: install

image: python:3.11-slim

workingDir: $(workspaces.src.path)

env:

- name: HOME

value: /tekton/home # avoid /.docker perms noise

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

value: "1"

script: |

#!/bin/sh

set -eu

echo "FPRINT: [pip-install] py-reqs-ci $(date -Iseconds)"

echo "PWD=$(pwd)"

echo "Listing:"

ls -al

echo "pip at: $(which pip)"

if [ -f requirements.txt ]; then

echo "requirements.txt found -> installing"

pip install -r requirements.txt

else

echo "requirements.txt NOT found -> skipping install"

fi

# 3) (Optional) training step — leave for later

# - name: train

# runAfter: [pip-install]

# workspaces:

# - name: src

# workspace: source

# taskSpec:

# workspaces: [{ name: src }]

# steps:

# - name: run-training

# image: python:3.11-slim

# workingDir: $(workspaces.src.path)

# env:

# - name: HOME

# value: /tekton/home

# script: |

# #!/bin/sh

# set -eu

# echo "FPRINT: [train] py-reqs-ci $(date -Iseconds)"

# echo "Training would happen here."

**pipelinerun.yaml**

apiVersion: tekton.dev/v1

kind: PipelineRun

metadata:

generateName: py-reqs-ci-

spec:

pipelineRef:

name: py-reqs-ci

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