Pipeline.yaml

apiVersion: tekton.dev/v1

kind: Pipeline

metadata:

name: py-spark-train-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

description: Shared workspace for code, deps, and outputs

tasks:

# 1) Clone the repository 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 -euo pipefail

echo "FPRINT: [clone] $(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)"

find . -maxdepth 2 -type f | sort

# 2) Install Python requirements if present

- name: install-requirements

runAfter: [clone]

workspaces:

- name: src

workspace: source

taskSpec:

workspaces:

- name: src

steps:

- name: pip-install

image: python:3.11-slim

workingDir: $(workspaces.src.path)

script: |

#!/bin/sh

set -euo pipefail

echo "FPRINT: [install] $(date -Iseconds)"

if [ -f requirements.txt ]; then

python -V

pip install --upgrade pip

pip install -r requirements.txt

else

echo "No requirements.txt found, skipping install"

fi

# 3) Run Spark training

- name: train-model

runAfter: [install-requirements]

workspaces:

- name: src

workspace: source

taskSpec:

workspaces:

- name: src

steps:

- name: spark-train

# Use your cluster-approved Spark image if required

image: bitnami/spark:3.3.1

workingDir: $(workspaces.src.path)

script: |

#!/bin/bash

set -euo pipefail

echo "FPRINT: [train] $(date -Iseconds)"

echo "Starting model training…"

# Adjust paths/files to your repo structure

if [ ! -f model\_training.py ]; then

echo "ERROR: model\_training.py not found in $(pwd)" >&2

exit 1

fi

# If ml\_stuff.zip is optional, guard it similarly

spark-submit \

--py-files ml\_stuff.zip \

model\_training.py

echo "Training finished."

pipelinerun.yaml

apiVersion: tekton.dev/v1

kind: PipelineRun

metadata:

generateName: py-spark-train-run-

spec:

pipelineRef:

name: py-spark-train-pipeline

# Optional: shorten or extend as you wish

timeouts:

pipeline: 45m

params:

- name: repo\_url

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

- name: revision

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

workspaces:

- name: source

volumeClaimTemplate:

spec:

accessModes: ["ReadWriteOnce"]

resources:

requests:

storage: 2Gi