**Pipeline.yaml**

apiVersion: tekton.dev/v1

kind: Pipeline

metadata:

name: basic-train

spec:

params:

- name: python\_image

type: string

# from your screenshot

default: docker-enterprise-prod.artifactrepository.citigroup.net/developersvcs-python-a1/miniconda-rhel8/23.5-py3.12:latest

- name: spark\_image

type: string

description: Internal Spark image that includes spark-submit

- name: main\_py

type: string

default: model\_training.py

- name: py\_files

type: string

default: "" # e.g., "ml\_stuff.zip" if you use it

workspaces:

- name: source

tasks:

- name: train

workspaces:

- name: src

workspace: source

taskSpec:

workspaces:

- name: src

steps:

# 1) Install requirements

- name: install

image: $(params.python\_image)

workingDir: $(workspaces.src.path)

script: |

#!/bin/sh

set -eu

if [ -f requirements.txt ]; then

pip install --no-cache-dir -r requirements.txt

else

echo "No requirements.txt; skipping install"

fi

# 2) Run training

- name: train

image: $(params.spark\_image)

workingDir: $(workspaces.src.path)

script: |

#!/bin/sh

set -eu

test -f "$(params.main\_py)" || { echo "Missing $(params.main\_py)"; exit 1; }

# Optional --py-files (only if the files exist)

PYFILES\_ARG=""

if [ -n "$(params.py\_files)" ]; then

IFS=',' read -r -a ARR <<<'$(params.py\_files)'

EXISTING=""

for f in "${ARR[@]}"; do

[ -f "$f" ] && EXISTING="${EXISTING:+$EXISTING,}$f"

done

[ -n "$EXISTING" ] && PYFILES\_ARG="--py-files $EXISTING"

fi

spark-submit $PYFILES\_ARG "$(params.main\_py)"

**pipelinerun.yaml**

apiVersion: tekton.dev/v1

kind: PipelineRun

metadata:

generateName: basic-train-run-

spec:

pipelineRef:

name: basic-train

params:

# Python is already defaulted to your miniconda image; override only if you want

# - name: python\_image

# value: docker-enterprise-prod.artifactrepository.citigroup.net/developersvcs-python-a1/miniconda-rhel8/23.5-py3.12:latest

- name: spark\_image

value: <PUT-YOUR-INTERNAL-SPARK-IMAGE-HERE> # must have spark-submit

- name: main\_py

value: model\_training.py

- name: py\_files

value: "" # or "ml\_stuff.zip"

workspaces:

- name: source

volumeClaimTemplate:

spec:

accessModes: ["ReadWriteOnce"]

resources:

requests:

storage: 1Gi