# CI/CD Pipeline in Azure DevOps Steps

**Scenario**

We want to build a CI/CD pipeline that:

1. Clones a GitHub or Azure Repos repository.
2. Installs Python dependencies.
3. Runs two scripts (fetch\_data.py and process\_data.py).
4. Publishes the processed output (processed\_data.json) as a pipeline artifact.

# Project Structure

azure-data-pipeline/ ├── data\_pipeline/

│ ├── fetch\_data.py

│ ├── process\_data.py

│ └── requirements.txt

└── azure-pipelines.yml

# Step 1: Python Scripts

## fetch\_data.py

import json def fetch():

data = {"students": [

{"id": 1, "name": "Abhinav", "marks": 78},

{"id": 2, "name": "Priya", "marks": 85},

{"id": 3, "name": "Rahul", "marks": 92},

]}

with open("raw\_data.json", "w") as f:

json.dump(data, f) print("Raw data fetched and saved to raw\_data.json")

if \_\_name\_\_ == "\_\_main\_\_":

fetch()

## process\_data.py

import json def process(): with open("raw\_data.json", "r") as f:

data = json.load(f) high\_scorers = [s for s in data["students"] if s["marks"] > 80] with open("processed\_data.json", "w") as f:

json.dump(high\_scorers, f) print("Processed data saved to processed\_data.json")

if \_\_name\_\_ == "\_\_main\_\_":

process()

## requirements.txt pandas

# Step 2: Azure Pipeline YAML

azure-pipelines.yml trigger:

* main # Runs pipeline when code is pushed to 'main' branch

pool: vmImage: 'ubuntu-latest'

steps:

# Step 1: Checkout code from repo

* task: Checkout@1

# Step 2: Set up Python - task: UsePythonVersion@0

inputs:

versionSpec: '3.10' addToPath: true

# Step 3: Install dependencies

* script: |

python -m pip install --upgrade pip pip install -r data\_pipeline/requirements.txt displayName: 'Install dependencies'

# Step 4: Fetch raw data

* script: | cd data\_pipeline python fetch\_data.py displayName: 'Fetch raw data'

# Step 5: Process data

* script: | cd data\_pipeline python process\_data.py displayName: 'Process data'

# Step 6: Publish output artifact - task: PublishBuildArtifacts@1

inputs:

PathtoPublish: 'data\_pipeline/processed\_data.json' ArtifactName: 'ProcessedData'

publishLocation: 'Container'

# Step 3: Theoretical Steps to Run in Azure DevOps

Since you I have a subscription, just documenting the process:

1. **Login to Azure DevOps Portal** o Go to https://dev.azure.com o Sign in with your Microsoft account.
2. **Create a New Project** o Click New Project. o Give a project name (e.g., AzureDataPipeline). o Choose visibility (Private or Public).
   * Click Create.
3. **Import Repository** o Inside your project → Go to Repos. o Import your GitHub/Azure Repos code into azure-data-pipeline folder.
4. **Create a Pipeline** o Go to Pipelines → New Pipeline. o Select your code source (GitHub or Azure Repos).
   * When asked for configuration, select YAML.
5. **Select YAML File** o Point to azure-pipelines.yml inside the root directory.
   * Azure DevOps automatically reads the YAML pipeline.
6. **Save and Run** o Click Save and Run.
   * Azure DevOps will:
     + Checkout repo.
     + Set up Python.
     + Install dependencies.
     + Run fetch\_data.py.
     + Run process\_data.py.
     + Save processed\_data.json as an artifact.
7. **View Pipeline Run** o In Pipelines → Click on the latest run.
   * Check logs for each step.
8. **Access Published Artifact** o In the pipeline run summary → Find Artifacts section.
   * Download ProcessedData → Inside, you’ll find processed\_data.json.

# Step 4: Final Outcome

* raw\_data.json (temporary raw data file).
* processed\_data.json (final processed file stored as artifact).
* This file can be later used in other pipelines or shared with teams.