CI/CD Pipeline in Azure DevOps – Theoretical 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

```yaml
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**
    - Go to https://dev.azure.com
    - Sign in with your Microsoft account.
2. **Create a New Project**
    - Click New Project.
    - Give a project name (e.g., AzureDataPipeline).
    - Choose visibility (Private or Public).
    - Click Create.
3. **Import Repository**
    - Inside your project → Go to Repos.
    - Import your GitHub/Azure Repos code into azure-data-pipeline folder.

4. **Create a Pipeline**
    - Go to Pipelines → New Pipeline.
    - Select your code source (GitHub or Azure Repos).
    - When asked for configuration, select YAML.
5. **Select YAML File**
    - Point to azure-pipelines.yml inside the root directory.
    - Azure DevOps automatically reads the YAML pipeline.
6. **Save and Run**
    - 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**
    - In Pipelines → Click on the latest run.
    - Check logs for each step.
8. **Access Published Artifact**
    - 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.