# **Week 5: Deploy & Automate with Azure DevOps**

### **Objective**

Automate the execution of the Supply Chain project scripts using Azure DevOps pipelines.

## **Step 1: Prepare the Project**

1. Organize your project folder to include all scripts and datasets from Weeks 1–4.

Create a requirements.txt file listing all the Python dependencies required. Example:  
  
 pandas

numpy

pyspark

1. Create a run\_pipeline.py script. This script should import and run the necessary project logic from your previous weeks.

## **Step 2: Push Code to Repository**

1. Log in to Azure DevOps.
2. Create a new project in Azure DevOps.
3. Push your local project folder to either **Azure Repos Git** or connect your repository from **GitHub**.

## **Step 3: Create a Pipeline in Azure DevOps**

1. In your Azure DevOps project, go to **Pipelines → New Pipeline**.
2. Choose the code source (Azure Repos Git or GitHub).
3. Select your repository.
4. Azure DevOps will suggest a starter pipeline. Replace it with your custom YAML.

## **Step 4: Define the Pipeline YAML**

Create a file named azure-pipelines.yml at the root of your repository with the following content:

trigger:

- main # Run pipeline when changes are pushed to the main branch

pool:

vmImage: 'ubuntu-latest' # Use Ubuntu environment

steps:

- script: |

pip install -r requirements.txt

python run\_pipeline.py

displayName: 'Run Supply Chain Script'

## **Step 5: Execute the Pipeline**

1. Commit and push the azure-pipelines.yml file to your repository.
2. Go to **Pipelines** in Azure DevOps and run the pipeline.
3. The pipeline will:  
   * Set up the virtual machine environment.
   * Install dependencies from requirements.txt.
   * Execute the run\_pipeline.py script.
   * Log the results in the console output.

## **Deliverables**

1. azure-pipelines.yml file defining the pipeline.
2. Execution log or console output from Azure DevOps after running the pipeline.