WEEK-5 PIPELINE AUTOMATIONS WITH AZURE DEVOPS

Step 1 — Prerequisites

- Azure DevOps account: https://dev.azure.com
- Source code in **Azure Repos** or GitHub
- ETL scripts, requirements.txt, and CSV generation code ready
- Azure Storage or Web App for deployment (optional but recommended)

Step 2 — Connect Your Git Repo

1. Go to **Azure DevOps → Repos → Import** or **push local repo** to Azure.
2. Make sure your **Python ETL scripts** (etl.py or data_collection_cleanup.py) and **requirements.txt** are committed.
3. Set up **two branches**:
    - dev → Testing
    - main → Production

Step 3 — Create Service Connection

1. Go to **Project Settings → Service Connections**
2. Click **New Service Connection → Azure Resource Manager**
3. Choose **Automatic Service Principal**
4. Select your subscription + resource group
5. Save it as → **Retail-Sales-Connection**

Step 4 — Create Variables

- Go to **Pipelines → Library → Variable Groups**
- Add variables like:

Variable	Example Value	Secret
DB_HOST	127.0.0.1	No
DB_USER	root	No
DB_PASSWORD	********	Yes
STORAGE_ACCOUNT	retailstorage	No
CONTAINER_NAME	salesdata	No

Step 5 — Create CI/CD Pipeline with YAML

Create a new file in your repo root: **azure-pipelines.yml**

```
trigger:
  branches:
    include:
      - main
      - dev

pool:
  vmImage: 'ubuntu-latest'

variables:
  pythonVersion: '3.10'
  outputFolder: 'outputs'

steps:
# Step 1 - Checkout Code
- task: Checkout@1
  displayName: 'Checkout source code'

# Step 2 - Setup Python
- task: UsePythonVersion@0
```

```
  inputs:
    versionSpec: '$(pythonVersion)'
    addToPath: true

# Step 3 - Install Dependencies
- script: |
    python -m pip install --upgrade pip
    pip install -r requirements.txt
  displayName: 'Install Python dependencies'

# Step 4 - Run ETL Scripts
- script: |
    echo "Running Retail Sales ETL Process..."
    python week2/Data_Collection_Cleanup.py
  displayName: 'Execute ETL Script'

# Step 5 - Publish Artifacts (Cleaned Data & Reports)
- task: PublishBuildArtifacts@1
  inputs:
    PathtoPublish: '$(System.DefaultWorkingDirectory)/cleaned_sales_data.csv'
    ArtifactName: 'CleanedData'
    publishLocation: 'Container'
  displayName: 'Publish ETL Outputs'

# Step 6 - Deploy to Azure Blob (CD)
- task: AzureCLI@2
  inputs:
    azureSubscription: 'Retail-Sales-Connection'
    scriptType: bash
```

```
  scriptLocation: inlineScript
  inlineScript: |
    echo "Uploading cleaned data to Azure Blob..."
    az storage blob upload \
      --account-name $(STORAGE_ACCOUNT) \
      --container-name $(CONTAINER_NAME) \
      --file $(System.DefaultWorkingDirectory)/cleaned_sales_data.csv \
      --name cleaned_sales_data.csv
displayName: 'Deploy to Azure Blob'
```

Step 6 — Pipeline Flow

1. **Trigger:** On commit to main or dev
2. **Install:** Python + dependencies
3. **Run ETL:** Executes scripts, generates cleaned_sales_data.csv
4. **Publish:** Saves artifacts for dashboards
5. **Deploy:** Pushes CSV to **Azure Blob** for Power BI or dashboards

Step 7 — Week 5 Deliverables

- **ETL Automation** → Auto-generated cleaned CSVs
- **CI Pipeline** → Build, test, artifacts
- **CD Pipeline** → Deploy results
- **Artifacts Produced:**
    - cleaned_sales_data.csv
    - category_metrics.csv
    - top3_products.csv
    - monthly_revenue.csv
    - underperforming_products.csv