Week 5: Deploy & Automate using Azure DevOps

Tools: Azure DevOps, Python, Git(Azure Repos)

Capstone Tasks Done:

Runs the Python ETL script (run_pipeline.py)

Cleans and processes supply chain data from sample_orders.json

Logs cleaned output to console

Saves it to cleaned_supply_chain_data.csv

Publishes the CSV as an artifact

Deliverables:

- Azure DevOps YAML pipeline
- Output log file

Step 1: Clone Azure DevOps Repository

1. Open your Azure DevOps project (e.g.,Supply Chain Monitoring)
2. Navigate to **Repos > Clone**
3. Use SSH or HTTPS URL.

```
git clone https://dev.azure.com/simantinip04/SupplyChainMonitoring/_git/SupplyChainMonitoring

cd SupplyChainMonitoring
```

Step 2: Add Week 5 Files

- Add the following to your cloned repo:

File	Purpose
run_pipeline.py	Calls your ETL logic
data_processing.py	Contains Week 2 data processing code
sample_orders.json	Sample input data
azure-pipelines.yml	Defines your CI pipeline

Step 3: Git Setup & Push Code

```
cd Week5
git config --global user.name "Simantini Patil"
git config --global user.email "simantini.p04@gmail.com"
git add .
git commit -m "Added Week 5 pipeline and ETL code"
git branch -M main
git push -u origin main
```

Step 4: Create and Run Pipeline

1. Go to Azure DevOps
2. Navigate to Pipelines > New Pipeline
3. Select:
    - Azure Repos Git
    - Choose your repo
    - Select "Existing Azure Pipelines YAML file"
    - Choose azure-pipelines.yml
4. Click Run Pipeline

Step 5: View Output

- After pipeline execution:
    - View logs in the DevOps console
    - The ETL process prints:
        - Raw data
        - Cleaned & processed data
        - Confirmation of CSV generation
    - Navigate to Artifacts tab
        - Download cleaned_supply_chain_data.csv

Result:

The pipeline completes successfully:

- Executes run_pipeline.py
- Cleans the data from sample_orders.json
- Generates cleaned_supply_chain_data.csv
- Publishes it as a downloadable artifact
- Logs status messages in the DevOps console