# Capstone Project: Supply Chain Monitoring and Optimization Platform

# Week 5 – Deploy & Automate with Azure DevOps

**Tools: Azure DevOps**

**Capstone Tasks:**

* Create a simple Azure DevOps pipeline to run Python scripts
* Install dependencies and execute the project
* Log the results and mark completion

## Objective

The goal of this stage is to automate the execution of our supply chain monitoring and optimization scripts using Azure DevOps Pipelines. This ensures repeatability, consistency, and reduces manual effort in running Python jobs.

## Step 1: Create an Azure DevOps Organization

1. Go to <https://azure.microsoft.com/en-us/products/devops>.  
2. Sign in with your Microsoft account.  
3. Create a new Organization (give it a unique name).  
4. Choose the region closest to you for better performance.

## Step 2: Create a New Project

1. Once inside your organization, click New Project.  
2. Enter:  
 - Project Name: SupplyChainAutomation  
 - Visibility: Public (recommended for free minutes)  
3. Click Create.

## Step 3: Push Code to Azure Repos (or connect GitHub)

1. Navigate to Repos in your project.  
2. Initialize a repo or connect to GitHub.  
3. Push your Python project files using Git commands:  
 git init  
 git remote add origin <repo-URL>  
 git add .  
 git commit -m 'Initial commit'  
 git push -u origin main

* **Create a simple Azure DevOps pipeline to run Python scripts (Step 4 and Step 5)**

## Step 4: Create a New Pipeline

1. Go to Pipelines → Create Pipeline.  
2. Select the repository.  
3. Choose YAML pipeline option.  
4. Use an existing azure-pipelines.yml file or create a new one.

* **Install dependencies and execute the project (Step 5)**

## Step 5: Define the Pipeline in YAML

Create a file named azure-pipelines.yml with the following content:

trigger:  
- main  
  
pool:  
 vmImage: 'ubuntu-latest'  
  
steps:  
- task: UsePythonVersion@0  
 inputs:  
 versionSpec: '3.x'  
 addToPath: true  
  
- script:   
 python -m pip install --upgrade pip  
 pip install -r requirements.txt  
 python run\_pipeline.py  
 displayName: 'Run Supply Chain Script'

* **Log the results and mark completion (Step 6 and Step 7)**

## Step 6: Save & Run the Pipeline

1. Commit the YAML file to the repository.  
2. Go back to Pipelines in Azure DevOps.  
3. The pipeline will be detected and start running.  
4. Logs will display Python setup, dependency installation, execution, and completion.

## Step 7: Verify Logs and Results

After completion, check the Job Summary:  
- Pipeline Status: Success/Failed  
- Execution logs of your Python program  
- Duration of pipeline run  
If failed, review logs to fix errors.

## Step 8: Automate with Scheduled Runs (Optional)

1. Go to Pipeline → Triggers → Scheduled.  
2. Add a CRON schedule (e.g., run daily at 6 PM).  
3. Ensures automatic execution at fixed intervals.

## Benefits of Using Azure DevOps for Supply Chain Monitoring

1. Automation – No manual execution of scripts.  
2. Scalability – Handles large datasets via cloud agents.  
3. Version Control Integration – Runs when new code is pushed.  
4. Reliability – Consistent execution with logs.  
5. Flexibility – Can integrate with Databricks, Spark, APIs, etc.