

# Week 5 – Automation via Azure DevOps

## Step 1: Create a New DevOps Project:

### 1. New Project -> Insert Name -> Create

The screenshot shows the Azure DevOps interface for a project named 'Data Engineering'. The left sidebar contains navigation links: Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area is titled 'Data Engineering' and includes an 'About this project' section with a description and an 'Add Project Description' button. To the right, there are 'Project stats' for the last 7 days, showing 0 pull requests and 86 commits by 2 authors. Below the stats is a 'Members' section showing one member, 'AS'.

## Step 2: Import repo into Azure DevOps

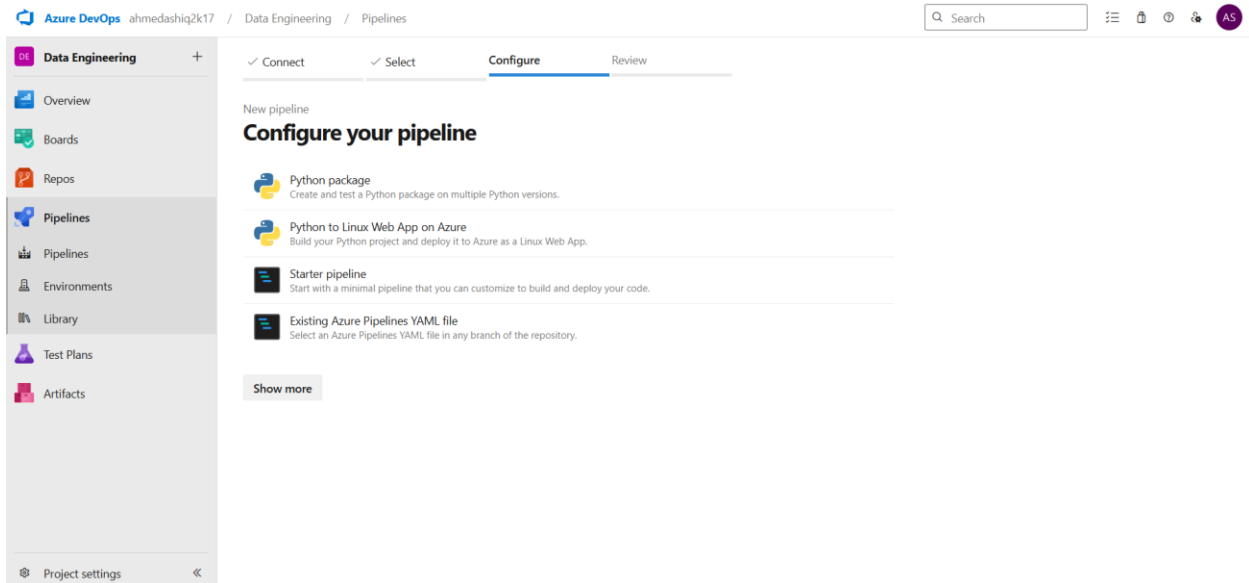
### 1. Repos -> Import -> Insert Git link -> Import

The screenshot shows the Azure DevOps interface for the 'Data Engineering' repository. The left sidebar contains navigation links: Overview, Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Advanced Security, Pipelines, Test Plans, Artifacts, and Project settings. The main content area is titled 'Data Engineering' and shows a list of commits. The table below contains the commit data:

Name ↑	Last change	Commits
Capstone Projects	Wednesday	cf244c4a Delete Capstone Projects/retail_sales_da...
June 03	Jun 3	f30548fa Delete June 03/large_employee_dataset...
June 04	Jun 4	7bb89fe0 Add files via upload ahmahmmmedd
June 09	Jun 9	1d627d2b Add files via upload ahmahmmmedd
June 10	Jun 10	f9d0318f Add files via upload ahmahmmmedd
June 11	Jun 11	991277a2 Add files via upload ahmahmmmedd
June 12	Jun 12	e134440f Add files via upload ahmahmmmedd
June 13	Jun 13	0f4fc0ec Add files via upload ahmahmmmedd
June 16	Jun 16	a9077229 Add files via upload ahmahmmmedd
June 17 and 18	Jun 18	26015447 Add files via upload ahmahmmmedd
June 19	Jun 19	48588437 Add files via upload ahmahmmmedd
June 20	Jun 25	647a0e25 Add files via upload ahmahmmmedd
June 25	Jun 25	4214b64e Update 3_trigger_another_dag.py ahma...
June 26	Jun 26	a3805af8 Add files via upload ahmahmmmedd

### Step 3: Create Pipeline

#### 1. Pipelines → Create Pipeline -> Azure Repos Git -> Choose Repo -> Python Package



### Step 4: Insert the yaml file code:

trigger: none

schedules:

- cron: "0 0 \* \* Mon"

displayName: Weekly Monday run

branches:

include:

- main

always: true

pool:

vmImage: 'ubuntu-latest'

steps:

- task: UsePythonVersion@0

inputs:

versionSpec: '3.8'

addToPath: true

- script: pip install pandas numpy pyspark

displayName: 'Install Python dependencies'

- script: |

python process\_attendance.py

python generate\_report.py

displayName: 'Process data and generate report'

- task: PublishPipelineArtifact@1

inputs:

targetPath: 'reports'

artifact: 'weekly\_attendance\_report'

publishLocation: 'pipeline'

## Step 5: Run the yaml code

### 1. Validate and save -> Run

```
1 trigger: none
2
3 schedules:
4   - cron: "0 0 * * Mon"
5     displayName: Weekly Monday run
6     branches:
7       include:
8         - main
9     always: true
10
11 pool:
12   vmImage: 'ubuntu-latest'
13
14 steps:
15   - task: UsePythonVersion@0
16     inputs:
17       versionSpec: '3.8'
18       addToPath: true
19
20   - script: pip install pandas numpy pyspark
21     displayName: 'Install Python dependencies'
22
23   - script: |
24     python process_attendance.py
25     python generate_report.py
26     displayName: 'Process data and generate report'
27
```

## Step 6: Summary of Pipeline

Summary Code Coverage

Manually run by Ahmed Sherif

Repository and version  
 Data Engineering  
main cbfe0758

Time started and elapsed  
 Just now  
<1s

Related  
 0 work items  
 0 artifacts

Tests and coverage  
[Get started](#)

[View 85 changes](#)

## Capstone Tasks:

### - Set up a DevOps pipeline to automate weekly processing

schedules:

- cron: "0 0 \* \* Mon"

displayName: Weekly Monday run

branches:

include:

- main

always: true

## **- Schedule the pipeline to run every Monday**

schedules:

- cron: "0 0 \* \* Mon"

displayName: Weekly Monday run

branches:

include:

- main

always: true

## **- Output a report with top 5 absentees or lowest performing departments**

### **# 1. Top 5 Absentees**

```
top_absentees = cleaned_df.filter(col("Clock_out").isNull()) \  
    .groupBy("Employee_ID") \  
    .agg(count("*").alias("Absence_Days")) \  
    .orderBy(desc("Absence_Days")) \  
    .limit(5)  
top_absentees.show()
```

### **# 2. Lowest Performing Departments**

```
dept_performance = cleaned_df.groupBy("Department") \  
    .agg(avg("work_hours").alias("Avg_Hours"),  
         avg(when(col("Status") == "completed",  
1).otherwise(0)).alias("Completion_Rate")) \  
    .orderBy("Completion_Rate") \  
    .limit(5)  
dept_performance.show()
```