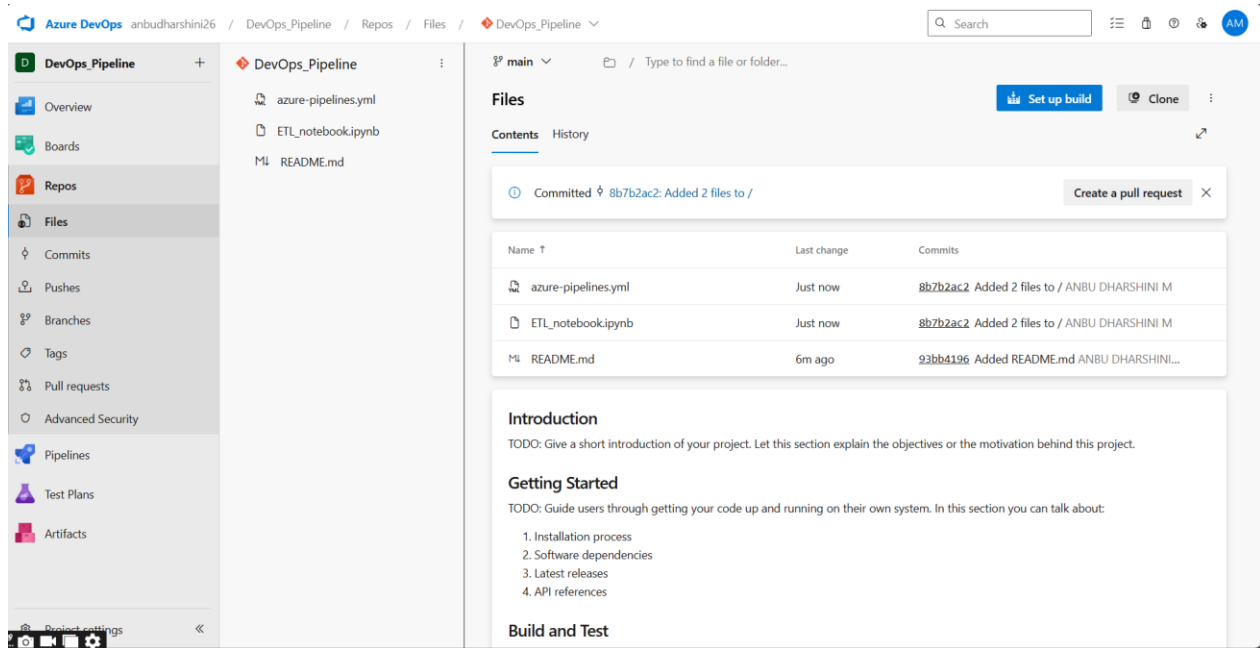


Pictorial Document for DevOps Pipeline Process – WEEK_5

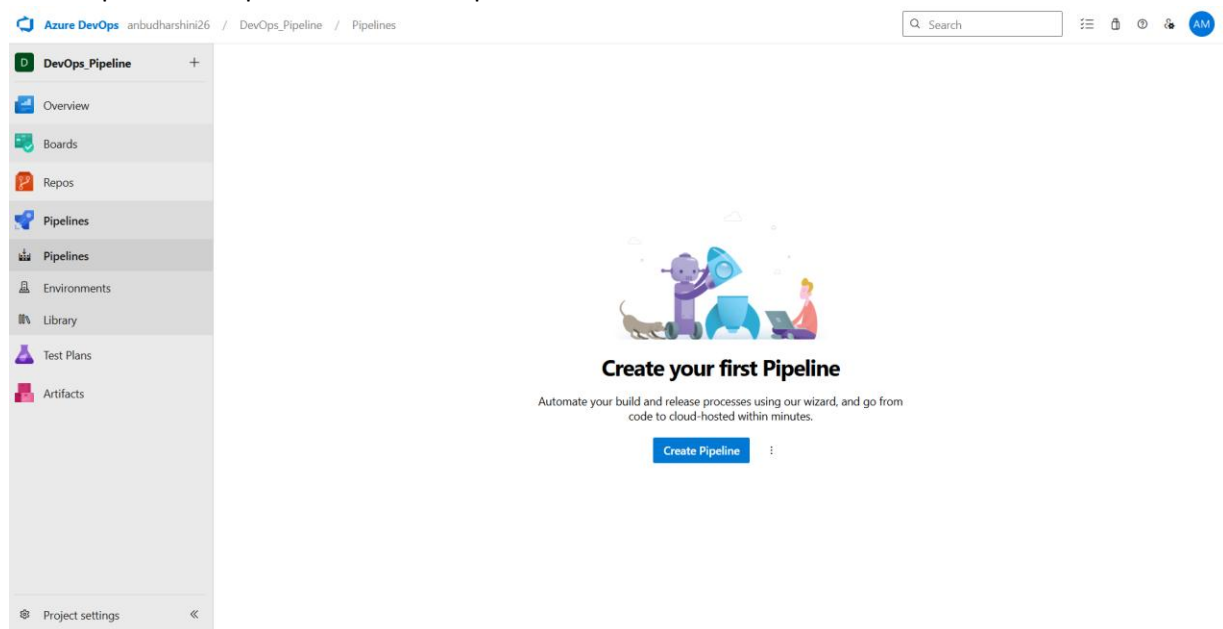
Step 1:

Create New Project -> SupplyChainAutomation -> Click on to the Three dots on the top right corner and add your .ipynb notebook -> “ETL_notebook.ipynb” and yaml file “azure-pipelines.yml” file.



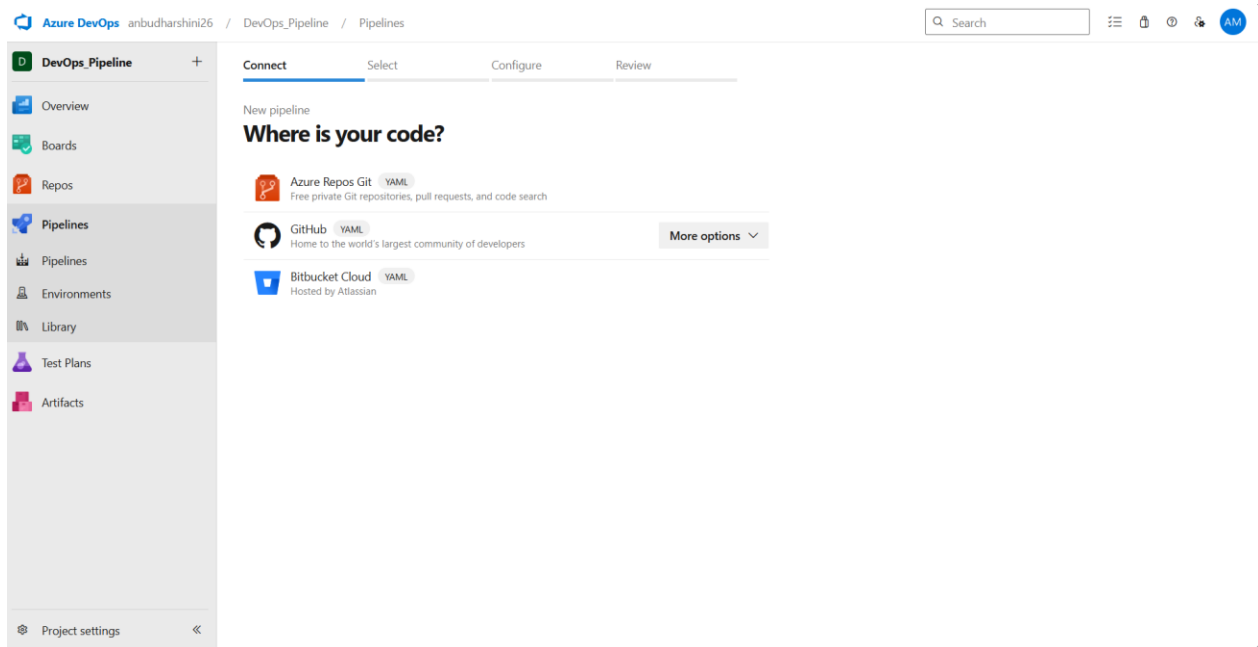
Step 2:

Go to Pipelines -> Pipelines -> Create Pipeline.



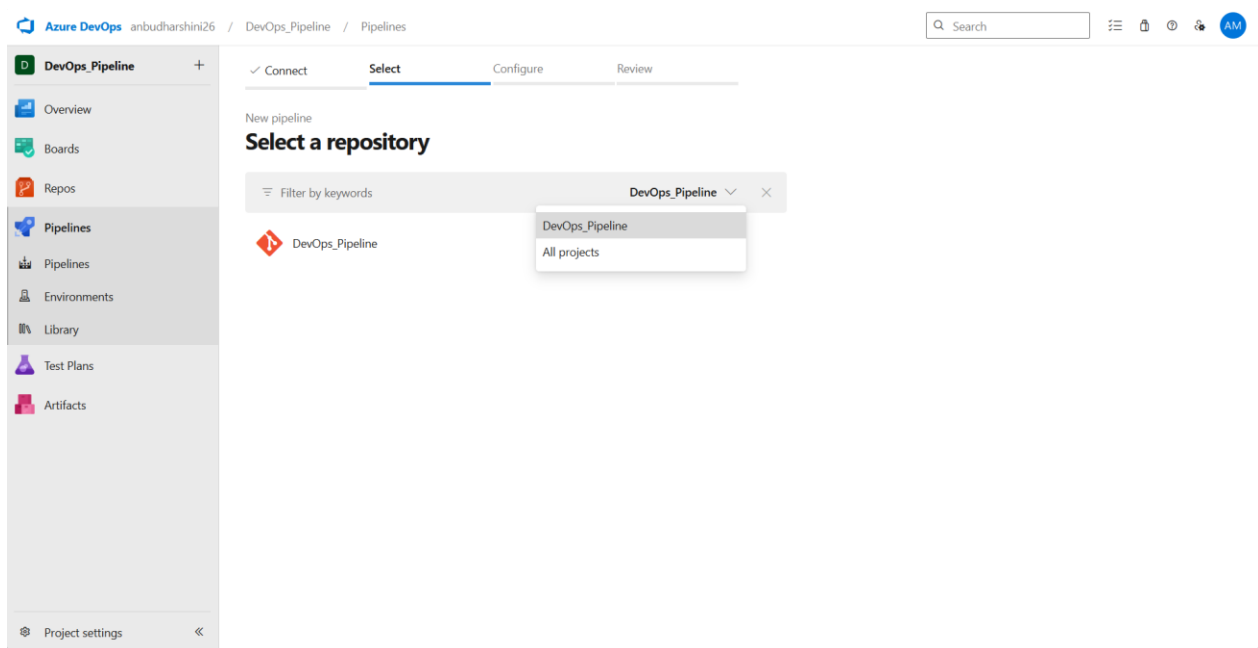
Step 3:

Select Azure Repos Git.



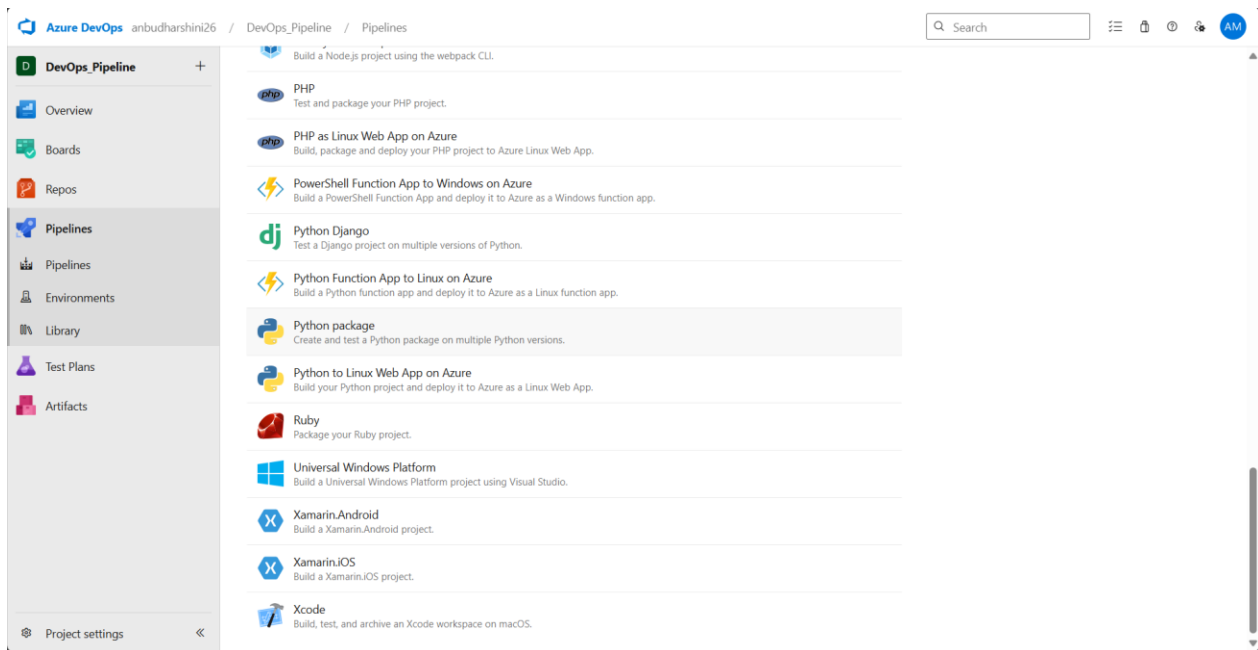
Step 4:

Select a repository -> DevOps_Pipeline.



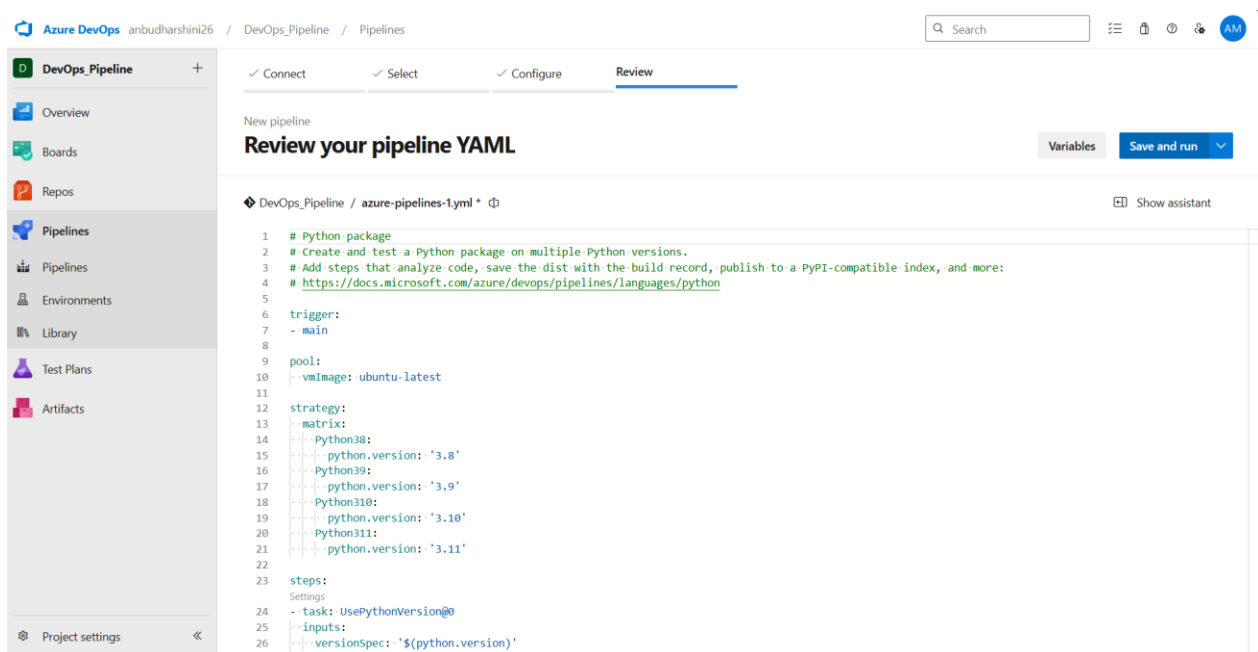
Step 5:

Select Python Package.



Step 6:

Click on Save and Run.



Step 7:

Again, click on Save and Run.

The screenshot shows the 'Review your pipeline YAML' dialog in Azure DevOps. The 'Review' tab is active, displaying the pipeline YAML code. A 'Save and run' modal is open on the right, prompting the user to commit the pipeline to the repository. The modal includes a 'Commit message' field with the text 'Set up CI with Azure Pipelines', an 'Optional extended description' field, and two radio button options: 'Commit directly to the main branch' (selected) and 'Create a new branch for this commit'. A 'Save and run' button is at the bottom right of the modal.

```
1 # Python package
2 # Create and test a Python package on multiple Python versions.
3 # Add steps that analyze code, save the dist with the build record, publish to a PyPI-com
4 # https://docs.microsoft.com/azure/devops/pipelines/languages/python
5
6 trigger:
7   - main
8
9 pool:
10  vmImage: ubuntu-latest
11
12 strategy:
13   matrix:
14     Python38:
15       python.version: '3.8'
16     Python39:
17       python.version: '3.9'
18     Python310:
19       python.version: '3.10'
20     Python311:
21       python.version: '3.11'
22
23 steps:
24   - task: UsePythonVersion@0
25     inputs:
26       versionSpec: '$(python.version)'
```

Step 8:

Now we can see the summary of the pipeline and it is scheduled to run with the configured agent.

The screenshot shows the pipeline summary page in Azure DevOps. The pipeline is titled '#20250822.1 • Set up CI with Azure Pipelines' and is owned by 'ANBU DHARSHINI M'. The 'Summary' tab is active, showing the repository and version, time started and elapsed, related work items, and tests and coverage. A 'Jobs' table is displayed below, showing a single job named 'Job' with a status of 'Queued'.

Name	Status	Duration
Job	Queued	

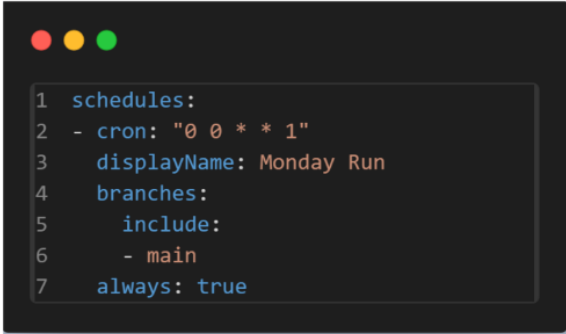
Capstone Tasks

1. Set up a DevOps pipeline to automate weekly processing

Entire workflow defines how to setup a devops pipeline and automate it.

2. Schedule the pipeline to run every Monday

During the YAML configuration in **step-5** we can define the schedule using cron expression




```
1 schedules:
2   - cron: "0 0 * * 1"
3     displayName: Monday Run
4     branches:
5       include:
6         - main
7     always: true
```

3. Output a report with top 5 absentees or lowest performing departments

The ETL_databricks_notebook.ipynb has the operations to report the

1. top 5 absentees



```
1 absenteesCount = dfEmp.groupBy("EmpID") \
2   .agg(
3     F.sum("IsAbsent").alias("AbsentCount")
4   ) \
5   .sort("AbsentCount", ascending=False) \
6   .limit(5)
7
8 absenteesCount.write.mode("overwrite").csv("abs_emp_count")
```

2. lowest performing departments

```
1 deptMetrics = dfJoined.groupBy("department") \  
2   .agg(  
3     F.sum("isLate").alias("LateCount"),  
4     F.sum("isAbsent").alias("AbsentCount"),  
5     F.round(F.mean("hoursWorked"), 2).alias("AverageWorkHours"),  
6     F.round(F.mean("productivityScore"), 2).alias("AverageProductivityScore"),  
7     F.sum("tasksCompleted").alias("TotalTasksCompleted")  
8   ) \  
9   .sort("AverageProductivityScore", ascending=True) \  
10  .limit(2)  
11  
12 deptMetrics.write.mode("overwrite").csv("deptMetrics")
```

Deliverables

YAML file and report file of latest attendance metrics is present in /Deliverables folder pushed into github.